

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Міністерство освіти і науки України

Кваліфікаційна наукова праця  
на правах рукопису

СУЛЕМА ЄВГЕНІЯ СТАНІСЛАВІВНА

УДК 004.62 : 004.043

ДИСЕРТАЦІЯ  
**МЕТОДИ, МОДЕЛІ ТА ЗАСОБИ ОБРОБКИ МУЛЬТИМОДАЛЬНИХ  
ДАНИХ ЦИФРОВИХ ДВІЙНИКІВ ДОСЛІДЖУВАНИХ ОБ'ЄКТІВ**

Спеціальність 01.05.03 – математичне та програмне забезпечення  
обчислювальних машин і систем

Подається на здобуття наукового ступеня доктора технічних наук

Дисертація містить результати власних досліджень.  
Використання ідей, результатів і текстів інших авторів  
мають посилання на відповідне джерело

\_\_\_\_\_ Є. С. Сулема

Науковий консультант

Дичка Іван Андрійович,  
доктор технічних наук, професор

Київ – 2020

## АНОТАЦІЯ

*Сулема Є. С.* Методи, моделі та засоби обробки мультимодальних даних цифрових двійників досліджуваних об'єктів. На правах рукопису.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 01.05.03 – математичне та програмне забезпечення обчислювальних машин і систем. Робота виконана на кафедрі програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» Міністерства освіти і науки України. Захист відбудеться у спеціалізованій вченій раді Д 26.002.02 при Національному технічному університеті України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, 2020.

Дисертацію присвячено вирішенню актуальної науково-технічної проблеми – підвищення ефективності обробки темпоральних мультимодальних даних в комп'ютерних системах.

В рамках дисертаційного дослідження задача обробки темпоральних мультимодальних даних в комп'ютерних системах розглядається в контексті створення теоретичних основ та практичних інструментів для розроблення нового класу програмного забезпечення систем обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів. Актуальність цього напрямку досліджень пояснюється необхідністю подальшої інформатизації суспільства та застосування людино-центрованого підходу, що ґрунтується на поєднанні технології цифрових двійників і технології мультимедіа для вирішення широкого кола задач у ситуаціях, пов'язаних з ризиком або обмеженням можливості безпосередньої участі людини. Необхідність цього нового підходу наочно продемонстрована багатьма техногенними та екологічними катастрофами, що трапились протягом останніх десятиліть. Прикладами таких катастроф

є аварії на ЧАЕС, АЕС Фукусіма-1 (Японія), заводі хімічних відходів м. Кельце (Польща) тощо. Ці та інші подібні ситуації, а також пандемія COVID-2019 свідчать, що в деяких випадках є необхідність у дистанційному форматі участі людини у вирішенні виробничих задач. При цьому в багатьох випадках є потреба у наданні людині (оператору, керівнику, користувачу тощо) всебічної інформації, що сприймається органами чуття людини, для забезпечення повноцінного заглиблення в ситуацію. Тому розроблення нового класу програмно-апаратних систем на основі технології цифрових двійників, технології мультимедіа, імерсійних технологій дозволить розширити можливості та підвищити ефективність людської діяльності в складних або нестандартних умовах.

Оскільки цифровий двійник характеризується не лише технічними характеристиками або поведінковими даними, але й візуальною моделлю, то доцільно розвивати технологію цифрових двійників у взаємозв'язку з технологією мультимедіа, яка передбачає оперування не лише аудіовізуальними даними, але й даними інших модальностей, які передають інші типи інформації, що сприймається органами чуття людини. Це дозволить синергетично посилити можливості обох технологій та сприятиме розробленню програмно-апаратних систем нового класу для вирішення людиною широкого кола задач у дистанційному форматі.

Незважаючи на наявність достатньої кількості науково-технічних рішень на основі технології цифрових двійників, на сьогодні відсутній системний підхід до вирішення задачі створення цифрових двійників, який враховував би всі аспекти застосування цієї технології: від розроблення моделей та оброблення мультимодальних даних цифрового двійника до забезпечення конфіденційності інформації, яка характеризує досліджуваний об'єкт. Тому в роботі пропонується комплексний підхід до вирішення задачі розроблення програмного забезпечення систем оброблення темпоральних мультимодальних даних цифрових двійників

досліджуваних об'єктів.

*Об'єктом дослідження* є процеси обробки темпоральних мультимодальних даних.

*Предметом дослідження* є методи обробки та моделі подання темпоральних мультимодальних даних у комп'ютерних системах.

*Метою дисертаційної роботи* є підвищення ефективності обробки темпоральних мультимодальних даних у комп'ютерних системах.

Результати дослідження викладено у п'яти розділах дисертації.

У розділі 1 проаналізовано проблему подання та обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів. Проаналізовано технологію цифрових двійників та програмне забезпечення, яке реалізує цю технологію для вирішення прикладних задач. Запропоновано людино-центрований підхід до відображення цифрових двійників за допомогою технології мультимедіа. Проаналізовано існуюче програмне забезпечення для створення мультимедійних застосунків та проблеми, пов'язані з його застосуванням для подання та обробки даних цифрових двійників. Запропоновані шляхи вирішення проблем подання та обробки даних цифрових двійників досліджуваних об'єктів.

У розділі 2 запропоновано теоретичні основи подання темпоральних мультимодальних даних цифрового двійника для опису досліджуваного об'єкта. Як інструмент такого опису розроблено апарат алгебраїчної системи агрегатів. Характерною рисою запропонованої алгебраїчної системи є врахування порядку слідування елементів при виконанні операцій та введенням операцій впорядкування і спеціальних відношень (частотних, інтервальних) між елементами. Визначено операції та відношення між елементами запропонованої алгебраїчної системи, в тому числі нечітко визначеними. Запропоновано концепцію мультиобразу для формальної специфікації моделей цифрового двійника та опису досліджуваного об'єкта.

Розділ 3 присвячено розробленню моделей подання та методів обробки темпоральних мультимодальних даних цифрового двійника досліджуваного об'єкта. Зокрема, розроблено метод створення мультиобразу досліджуваного об'єкта, який полягає в послідовному виконанні процедур формування структури даних, декомпозиції специфікації мультиобразу, підготовки темпоральних мультимодальних даних, об'єднанні, сортуванні та проріджуванні даних. Розроблено метод синхронізації мультиобразів, який полягає в виконанні процедур формування шаблону синхронізації, впорядкування мультиобразів, попарної синхронізації мультиобразів. Розроблено метод адаптивного ущільнення графічних даних цифрового двійника, який ґрунтується на застосуванні морфологічних операцій та визначенні областей інтересу. Розроблено метод конфіденційного зберігання даних цифрового двійника, особливістю якого є застосування процедури забезпечення конфіденційності даних залежно від категорії даних. Розроблено три моделі цифрового двійника: муксельну модель, часову зв'язну модель та багаторівневу онтологічну модель. Ці моделі дозволяють представляти досліджуваний об'єкт на різних рівнях абстракції. Муксельна модель призначена для подання деталізованої інформації про об'єкт дослідження. Часова зв'язна модель слугує для відображення об'єкта у динаміці зміни його стану. У багаторівневій онтологічній моделі досліджуваний об'єкт подається як композиція його складових. Комбінування цих моделей дозволяє представити об'єкт дослідження максимально повно.

У розділі 4 представлено парадигму програмування мультиобразів для обробки темпоральних мультимодальних даних цифрових двійників, визначальною рисою якої є використання нової сутності – мультиобразу досліджуваного об'єкта як основного компонента обчислювального процесу, що дозволяє спростити технологію розроблення програмного забезпечення систем обробки темпоральних мультимодальних даних

цифрових двійників досліджуваних об'єктів. Особливостями цієї парадигми є застосування апарату алгебраїчної системи агрегатів для обробки темпоральних мультимодальних даних; використання синхронізації та агрегації даних різних модальностей як основної складової моделі обчислень; орієнтація на застосування різнорідних джерел даних. Для реалізації парадигми програмування мультиобразів запропоновано базову модель обчислень та модель обчислень для технології цифрових двійників. Розроблено мову програмування ASAMPL для обробки структур темпоральних мультимодальних даних, представлено її синтаксис та семантику, а також продемонстровано застосування мови програмування ASAMPL для вирішення прикладних задач.

Розділ 5 присвячено програмним системам для обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів. Зокрема, розроблено архітектуру програмної системи обробки темпоральних мультимодальних даних цифрових двійників та запропоновано варіанти її використання для розроблення програмного забезпечення для галузі охорони здоров'я та освіти. Особливістю розробленої архітектури є забезпечення виконання дворівневої процедури синхронізації та агрегації темпоральних мультимодальних даних, що дозволяє підвищити ефективність обробки темпоральних мультимодальних даних та спростити процес розроблення прикладного програмного забезпечення систем обробки темпоральних мультимодальних даних.

У додатках наведено інформацію про впровадження результатів дослідження (Додаток А) та інформацію про наукові публікації і апробацію отриманих результатів (Додаток Б).

**Ключові слова:** прикладне програмне забезпечення, парадигма програмування, мова програмування, модель досліджуваного об'єкта, темпоральні мультимодальні дані, цифровий двійник, мультимедіа.

## ABSTRACT

*Sulema Ye. S.* Methods, models, and means of researched objects digital twins multimodal data processing. On the rights of the Manuscript.

The thesis for the degree of Doctor of technical sciences, specialty 01.05.03 – Mathematical Support and Software of Computers and Systems. The work was performed at the Computer Systems Software Department of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute». The dissertation will be held at the specialized scientific council D 26.002.02 of the National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, 2020.

The dissertation is devoted to solving a topical scientific and technical problem which is the increasing of temporal multimodal data processing efficiency in computer systems.

In the dissertation research, the problem of temporal multimodal data processing in computer systems is considered in the context of creating theoretical and practical bases for the development of a new class of software for processing temporal multimodal data of digital twins of researched objects. This area of research is topical due to the need for further digitalization of human activities and the use of human-centred approach based on a combination of digital twins technology and mulsemmedia technology to solve a wide range of problems in situations involving risk or limiting direct human participation. The necessity for this new approach is clearly demonstrated by the many anthropogenic and environmental catastrophes which have occurred in recent decades. Examples of such catastrophes are the Chernobyl disaster, the Fukushima-1 nuclear power plant accident (Japan), the Kielce chemical waste plant accident (Poland), etc. These and other similar situations as well as the COVID-2019 pandemic indicate that in some cases there is a need for a remote format of human participation in solving work problems. At the same time, in many cases there is a need to provide a person (operator, manager, user, etc.) with

comprehensive information perceived by the human senses to ensure a full immersion in the situation. Therefore, the development of a new class of software and hardware systems based on digital twins technology, mulsemmedia technology, immersion technologies will expand the capabilities and increase the efficiency of human activities in complicated or non-standard conditions.

Since the digital twin is characterized not only by technical characteristics or behavioural data, but also by a visual model, it is advisable to develop digital twins technology in conjunction with mulsemmedia technology which involves operating not only audio-visual data, but also data of other modalities that represent other types of information perceived by the human senses. This will synergistically strengthen the capabilities of both technologies and promote the development of software and hardware systems of a new class to solve a wide range of problems in a remote format.

Despite the availability of a sufficient number of scientific and technical solutions based on digital twins technology, today there is no systematic approach to solving the problem of creating digital twins which would take into account all aspects of applying this technology: from model development and of multimodal digital twins data processing to ensuring confidentiality of information which characterizes the researched object. Therefore, the dissertation proposes a comprehensive approach to solving the problem of software development for systems of processing temporal multimodal data of digital twins of researched objects.

*The object of the research* is the processes of temporal multimodal data processing.

*The subject of the research* is the methods of processing and models of representation of temporal multimodal data in computer systems.

*The purpose of the dissertation* is to increase the efficiency of processing temporal multimodal data in computer systems.

The results of the research are presented in five sections of the dissertation.



Section 1 analyses the problem of presenting and processing temporal multimodal data of digital twins of researched objects. Both the digital twins technology and the software which implements this technology for solving applied problems are analysed. A human-centred approach to the representation of digital twins using mulsemmedia technology is proposed. Both existing software for creating mulsemmedia applications and problems related to its use for presenting and processing data of digital twins are analysed. The ways of solving the problems of presentation and data processing of digital twins of researched objects are proposed.

Section 2 presents the theoretical basis for representing the temporal multimodal data of a digital twin to describe the researched object. The apparatus of the algebraic system of aggregates is developed as a tool for such a description. A characteristic feature of the proposed algebraic system is consideration of the elements order when performing operations as well as the introduction of ordering operations and special relations (frequency, interval) between the elements. The operations and relations between the elements of the proposed algebraic system, including fuzzy elements, are defined. The concept of a multi-image is proposed for the formal specification of digital twin models and the description of the researched object.

Section 3 presents the models of representation and methods of processing the temporal multimodal data of a digital twin of the researched object. In particular, a method for creating multi-image of the researched object is developed; it consists in the sequential implementation of procedures for the data structure forming, the multi-image specification decomposition, temporal multimodal data preparing, data fusion, sorting, and singling. A method for multi-images synchronization is developed; it consists in the implementation of procedures for synchronization template forming, multi-images ordering, and pairwise synchronization of multi-images. A method for digital twin graphical data adaptive compression is developed; it is based on the application of

morphological operations and defining regions of interest. A method of confidential storage of digital twin data is developed; the distinguishing feature of this method is the application of the data protection procedure depending on the data category. Three digital twin models are developed: the muxel model, the temporal linked model, and the multilevel ontological model. These models allow to represent the object at different levels of abstraction. The muxel model is designed to provide detailed information about the researched object. The temporal linked model is used to represent the object in the dynamics of changes in its state. In a multilevel ontological model, the researched object is represented as a composition of its components. Combining these models enables representation of the researched object as completely as possible.

Section 4 presents the paradigm of multi-images programming for digital twins temporal multimodal data processing; its distinguishing feature is the use of a new entity – a multi-image of the researched object as the main component of the computational process that allows to simplify the technology of software development for systems of processing temporal multimodal data of digital twins of researched objects. Features of this paradigm are the use of the apparatus of the algebraic system of aggregates for processing temporal multimodal data; the use of data synchronization and aggregation of different modalities as the main component of the computational model; the focus on the use of heterogenous data sources. To implement the multi-image programming paradigm, a basic computational model and a computational model for digital twins technology are proposed. The ASAMPL programming language for processing temporal multimodal data structures is developed; its syntax and semantics are presented, and the application of the ASAMPL programming language for solving applied problems is demonstrated.

Section 5 deals with software systems for processing temporal multimodal data of digital twins of the researched objects. In particular, the architecture of the software system for processing temporal multimodal data of

digital twins is proposed and the options of its use for the development of software for the area of health care and distance education are demonstrated. A feature of the developed architecture is the use of a two-level procedure for synchronization and aggregation of temporal multimodal data. The proposed approach provides efficient processing temporal multimodal data as well as it enables both reducing programming man-hours and simplifying the process of applied software development for systems of processing temporal multimodal data.

The appendices provide information on the implementation of the research results (Appendix «А») and information on scientific publications and approbation of the obtained results (Appendix «Б»).

**Key words:** applied software, programming paradigm, programming language, researched object model, temporal multimodal data, digital twin, mulsemedia.

## **СПИСОК ПУБЛІКАЦІЙ ЗА ТЕМОЮ ДИСЕРТАЦІЙНОЇ РОБОТИ / THE LIST OF AUTHOR'S PUBLICATIONS FOR DISSERTATION THESES**

Безпосередньо за тематикою дисертаційної роботи опубліковано 72 наукові роботи, а саме: /

Directly for the subject of dissertation theses 72 scientific publications have been published:

*Наукові публікації, в яких висвітлені основні наукові результати дисертації (26) /*

*The main scientific results of the dissertation theses were published in the next publications (26):*

*- монографії (1) / monographs (1):*

1. Mathematical Methods in Interdisciplinary Sciences / Sulema Ye., Kerre E., et al.; editor Chakraverty S. Wiley, USA, 2020. 464 p. ISBN : 9781119585640. DOI : 10.1002/9781119585640.

**- статті в наукових фахових журналах України (19) /**

**- articles in Ukrainian professional journals (19):**

2. Sulema Ye.S., Rvach D.V. Models of computation for Digital Twins data processing. *Наукові вісті КНУ*, 2020. № 2, С. 74–81.
3. Sulema Ye. Multimodal data processing based on algebraic system of aggregates relations. *Radio Electronics, Computer Science, Control*, 2020. № 1, С. 169–180. [Категорія “А”].
4. Sulema Ye., Peschanskii V. Timewise data processing with programming language ASAMPL. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Технічні науки*, 2020. Т.31(70), Част.1. № 1, С. 132–137.
5. Дичка І.А., Сулема Є.С. Модель подання мультимодальних даних для комплексного опису об’єктів спостереження. *Вісник Вінницького політехнічного інституту*, 2020. № 1, С. 53–60.
6. Sulema Ye., Glinskii V. Semantics and pragmatics of programming language ASAMPL. *Проблеми програмування*, 2020. № 1, С. 74–83.
7. Sulema Ye., Rudenko С. A method of artifact compensation for dual quaternion skinning and its application in digital twin models. *Вісник Хмельницького національного університету. Технічні науки*, 2020. № 1, С. 12–19.
8. Сулема Є.С., Радченко Є.О. Метод стеганографічного захисту мультимедійних даних на основі процедури псевдовипадкового вбудовування. *Наукові вісті КНУ*, 2020. № 1, С. 40–47.
9. Sulema Y.S., Los I.A. Levels-Of-Detail generation method for skeletal meshes. *Системні технології*, 2019. Т.6. № 125, С. 3–14.
10. Сулема Є.С., Топчієв Б.С. Інтелектуальна колоризація зображень за допомогою генеративних змагальних мереж. *Системні технології*, 2019. Т.5. № 124. С. 94–103.
11. Dychka I., Sulema Ye., Bukhtiarov Iu. Digital Twin Information Technology for Biomedical Data Complex Representation and Processing. *Вісник*

*Херсонського національного технічного університету*, 2019. № 3 (70), С. 112–119.

- 12.Dychka I.A., Sulema Ye.S. Ordering Operations in Algebraic System of Aggregates for Multi-Image Data Processing. *Наукові вісті КПП*, 2019. № 1, С. 15–23.
- 13.Dychka I.A., Sulema Ye.S. Logical Operations in Algebraic System of Aggregates for Multimodal Data Representation and Processing. *Наукові вісті КПП*, 2018. № 6, С. 44–52.
- 14.Shkurat O.S., Sulema Ye.S., Dychka A.I. Complicated Shapes Estimation Method for Objects Analysis in Video Surveillance Systems. *Наукові вісті КПП*, 2018. № 3, С. 53–62. DOI : 10.20535/1810-0546.2018.3.136433.
- 15.Dychka I.A., Sulema Ye.S., Chernykh D.A. Rasterization Method for Voxel Model Cutting. *Наукові вісті КПП*, 2018. № 2, С. 25–32.
- 16.Dychka I., Shyrochyn S., Sulema Ye. Analysis of Parallel Computations Efficiency for User's Private Multimedia Data Protection in Clouds. *Наукові вісті КПП*, 2016. Т.1, С. 40–46.
- 17.Резнік К.Ю., Сулема Є.С. Метод локалізації вушної раковини на зображенні людини у профіль. *Вісник Хмельницького національного університету*, 2014. № 4, С. 113–119.
- 18.Сулема Є.С., Широчин С.С. Спосіб стеганографічного захисту даних в аудіо-файлах на основі комплементарного образу. *Наукові вісті КПП*, 2014. Т.61, С. 80–87.
- 19.Сулема Є.С., Широчин С.С. Спосіб стеганографії зображень на основі комплементарного образу. *Захист інформації*, 2013. Т.4, С. 345–353.
- 20.Сулема Є.С., Широчин С.С. Спосіб стеганографії зображень з фрагментацією стегоданих та розділенням закритого ключа. *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні*, 2012. Т.1(22), С. 64–68.

**- статті в закордонних фахових виданнях, які реферуються базою Scopus (6), в тому числі:**

**- articles in foreign professional journals referenced by the Scopus database (6), including:**

**- третього квартиля (Q3) (4) / the third quartile (Q3) (4):**

- 21.Sulema Ye. ASAMPL: Programming Language for Mulsemmedia Data Processing Based on Algebraic System of Aggregates. *Advances in Intelligent Systems and Computing*, 2018. Vol.725, P. 431–442. ISSN : 21945357.
- 22.Radchenko Ye., Dychka I., Sulema Ye., Suschuk-Sliusarenko V., Shkurat O. Steganographic Protection Method Based on Huffman Tree. *Advances in Intelligent Systems and Computing*, 2019. Vol.902, P. 283–292. ISSN : 21945357.
- 23.Hu Zh., Dychka I., Sulema Ye., Valchuk Yu., Shkurat O. Method of medical images similarity estimation based on feature analysis. *International Journal of Intelligent Systems and Applications (IJISA)*, 2018. Vol.10. No. 5, P. 14–22. ISSN : 20749058.
- 24.Hu Zh., Dychka I., Sulema Ye., Radchenko Ye. Graphical Data Steganographic Protection Method Based on Bits Correspondence Scheme. *International Journal of Intelligent Systems and Applications (IJISA)*, 2017. Vol.9. No.8, P.34–40. ISSN : 20749058.

**- четвертого квартиля (Q4) (1) / the fourth quartile (Q4) (1):**

- 25.Sulema Ye., Dychka I., Sulema O. Multimodal Data Representation Models for Virtual, Remote, and Mixed Laboratories Development. *Lecture Notes in Networks and Systems*, 2018. Vol.47, P. 559–569. ISSN : 23673389.

**- реферується базою Scopus (1) /**

**- referenced by the Scopus database (1):**

- 26.Yevgeniya Sulema, Etienne Kerre, Oksana Shkurat. Vector Image Retrieval Methods Based on Fuzzy Patterns. *International Journal of Modern Education and Computer Science (IJMECS)*. 2020. Vol.12. No.3. P. 8–16. DOI : 10.5815/ijmecs.2020.03.02.

**За результатами досліджень також опубліковано статті в інших наукових журналах України та за кордоном (7) /**

**According to the results of investigations, the articles in other scientific Ukrainian and foreign journals are also published (7):**

27. Sulema Ye., Papusha R. Method of Raster Line Dynamic Smoothing. *Геометричне моделювання та інформаційні технології*, 2017. № 2 (4), С. 84–89.
28. Сулема Є.С., Широчин С.С. Метод захисту зображень на основі шифрування палітри. *Вісник Хмельницького національного університету*, 2014. №3, С. 114–119.
29. Сулема Є.С., Ебрахімі Кахоу С. Адаптивний спосіб ущільнення зображень. *Вісник Хмельницького національного університету*, 2010. № 2, С. 125–131.
30. Павловець О.В., Сулема Є.С. Метод розпізнавання мовних одиниць на основі ADABOOST. *Вісник Хмельницького національного університету*, 2013. № 4, С. 211–222.
31. Sulema Ye., Ebrahimi Kahou S. Comparative Study of Image Compression Algorithms. *Radioelectronics and Informatics*, 2009. № 1. P. 1–8.
32. Сулема Є.С., Ебрахімі Кахоу С., Лавренчук С.С. Віртуальна мультимедійна лабораторія: особливості організації та обробки даних. *Збірник наукових праць Національного гірничого університету*, 2009. Vol.2. № 33. С. 90–96.
33. Sulema Ye., Cvjetkovic V. Remote Laboratory for Supporting e-Studies in Electronics. *International Journal of Online Engineering*, 2006. Vol 2, № 1, P. 1–9.

***Результати, викладені в дисертації, було апробовано на 31 науковій конференції, зокрема опубліковано 39 матеріалів конференцій /***

***The results shown in the theses were validated at 31 scientific conferences, in particular, were published in 39 materials of the conferences:***

***- що реферується базою Scopus (10) /***

***- which are referenced by the Scopus database (10):***

34. Sulema Ye., Amram N., Aleshchenko O., Sivak O. Quality of Experience Estimation for WebRTC-based Video Streaming. *Proceedings of the 24th*

- International Conference “European Wireless” (EW2018). Catania, Italy, 2018. P. 1–6.
35. Dychka I., Sulema Ye., Rudenko C. A Mathematical Model of Microsurface Normal Distribution for Specular Bidirectional Reflectance Distribution Function. Proceedings of the International Conference “Advanced Computer Information Technologies” (ACIT 2018). Ceske Budejovice, Czechia, 2018, P. 30–33.
  36. Sulema Ye., Rozinaj G. WebRTC-based 3D Videoconferencing System. Proceedings of 59th International Symposium ELMAR-2017. Zadar, Croatia, 2017, P. 193–196.
  37. Sulema Ye., Rozinaj G. Multimodal Information in Telecommunication Systems: Analysis of Technological Readiness. Proceedings of 58th International Symposium ELMAR-2016. Zadar, Croatia, 2016, P. 119–122.
  38. Sulema Ye. Mulsemmedia Vs. Multimedia: State of the Art and Future Trends (Invited Paper). Proceedings of the 23<sup>rd</sup> IEEE International Conference on Systems, Signals and Image Processing (IWSSIP2016). Bratislava, Slovakia, 2016, P. 19–23.
  39. Sulema Ye. Image Protection Method Based on Binary Operations. Proceedings of the 23<sup>rd</sup> IEEE International Conference on Systems, Signals and Image Processing (IWSSIP2016). Bratislava, Slovakia, 2016, P. 295–298.
  40. Sulema Ye. Haptic Interaction in Educational Applications. Proceedings of the International Conference on Interactive Mobile Communication, Technologies and Learning (IMCL2015). Thessaloniki, Greece, 2015, P. 312–314.
  41. Kovács P.T., Murray N., Rozinaj G., Sulema Ye., Rybárová R. Application of Immersive Technologies for Education: State of The Art. Proceedings of the International Conference on Interactive Mobile Communication, Technologies and Learning (IMCL2015). Thessaloniki, Greece, 2015, P. 283–288.
  42. Sulema Ye., Ebrahimi Kahou S. Statistical Approach to Image Classification. Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2010). Cluj-Napoca, Romania, 2010, Vol. 3, P. 1–4.



43. Sulema Ye., Ebrahimi Kahou S. Image Compression: Comparative Analysis of Basic Algorithms. Proceedings of the IEEE International Conference EWDTS'09. Moscow, Russia, 2009, P. 534–537.

*- інші конференції (29) / other conferences (29):*

44. Pester A., Sulema Ye. Multimodal Data Representation Based on Multi-Image Concept for Immersive Environments and Online Labs Development. Proceedings of the International Conference on Interactive Collaborative and Blended Learning (ICBL2019). Santiago de Cuba, Cuba, 2019. P. 1–18.
45. Dychka I., Sulema O., Salenko A., Sulema Ye. Augmented Reality Application Based on Information Barcoding. Proceedings of the International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2019). Thessaloniki, Greece, 2019. P. 1–12.
46. Sulema Ye., Rudenko C. Fast Approximated Subsurface Scattering. Proceedings of 12<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2018. Bratislava, Slovakia, 2018. P. 1–4.
47. Shkurat O., Sulema Ye., Suschuk-Sliusarenko V., Dychka A. Image Segmentation Method Based on Statistical Parameters of Homogeneous Data Set. Proceedings of International Conference of Artificial Intelligence, Medical Engineering, Education, 2018. 12 p.
48. Sulema Ye., Bartkoviak A., Sokolovska A., Kapura A., Ohoiko M., Hurov V. Smart Room Software Based on Microsoft Kinect and Speech Synthesizing. Proceedings of 11<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2017. Bratislava, Slovakia, 2017, P. 71–74.
49. Sulema Ye., Zaichuk V., Chernyk D. Voxel-to-Polygonal Model Conversion Method. Proceedings of 11<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2017. Bratislava, Slovakia, 2017, P. 67–70.
50. Радченко Є.О., Сулема Є.С. Спосіб стеганографічного захисту графічних даних на основі схеми відповідності бітів та аналізу візуальних властивостей контейнера. Матеріали доповідей Шостої Міжнародної науково-практичної конференції з сучасних проблем

- кодування, захисту й ущільнення інформації. Вінниця, Україна, 2017, С. 51–53.
51. Sulema Ye., Zaichuk V., Chernykh D. Haptic Interaction in 3D World with Use of Data Glove and Web-Camera. Proceedings of 10<sup>th</sup> International Workshop on Multimedia and Signal Processing Redžúr 2016. Slovakia, 2016, P. 45–48.
52. Sulema Y.S., Shkurat O.S. Information System for Archival Medical Images Automated Processing. Book of Abstracts of the 3rd International Conference Health Technology Management HTM-2016. Chisinau, Moldova, 2016, P. 72.
53. Sulema Ye., Valchuk Yu. Brain Tissue Image Adjustment Method. Book of Abstracts of the AMMODIT and final EUMLS Workshop. Germany, 2016, P. 31–32.
54. Sulema Ye., Shyrochyn S., Protection of Personal Graphic User Data Transmission in Computer Networks. Proceedings of International conference “Information Technology. Problems and Solutions”. Ufa, 2014, P. 7–10.
55. Сулема Є.С., Широчин С.С. Аналіз ефективності паралельної реалізації алгоритмів захисту зображень. Збірник наукових праць за матеріалами Восьмої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2014). Хмельницький, Україна, 2014, С. 335–342.
56. Сулема Є.С., Широчин С.С. Підвищення стійкості LSB-стегосистем шляхом аналізу і корекції характеристик контейнера. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький, Україна, 2012, С. 319–327.
57. Сулема Є.С., Гуренко О. Спосіб обмеження прав доступу до елементів зображення. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький, Україна, 2012, С. 309–313.
58. Сулема Є.С., Малашихіна А. Обробка аудіо-сигналів у віртуальній мультимедійній лабораторії. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький,

Україна, 2012, С. 314–318.

59. Сулема Є.С., Гнатейко В. Дослідження методів пошуку музичних файлів. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький, Україна, 2012, С. 328–332.
60. Сулема Є.С., Гуренко О. Модифікований алгоритм пошуку текстової інформації на зображеннях. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2011). Хмельницький, Україна, 2011, С. 202–206.
61. Сулема Є.С., Широчин С.С. Засоби подання інформації неграфічного характеру в картографічних зображеннях. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2011). Хмельницький, Україна, 2011, С. 207–213.
62. Сулема Є.С., Заворотній А. Спосіб розпознавання музикальних звукових образів. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2011). Хмельницький, Україна, 2011, С. 214–218.
63. Сулема Є.С., Широчин С.С. Критерії пошуку оптимального розташування блоків стеганографічних даних в контейнері. Матеріали Міжнародної науково-технічної конференції «Системний аналіз та інформаційні технології» (SAIT 2011). Київ, Україна, 2011, С. 516.
64. Сулема Е., Эбрахими Кахоу С. Пиксельно-объектный способ классификации изображений. Труды десятой международной научно-практической конференции «Современные информационные и электронные технологии». Одесса, 2009. Т.1, С. 133.
65. Сулема Є.С., Ебрахімі Кахоу С. Покращення якості ущільнених зображень в автоматизованих системах відео-спостереження. Матеріали XVI Міжнародної конференції з автоматичного управління «Автоматика-2009». Чернівці, Україна, 2009, С. 54.
66. Sulema Ye., Radchuk V. Audio Processing in the Virtual Multimedia Laboratory. Proceedings of the International Conference ICL2007. Villach, Austria, 2007. P. 1–5.
67. Sulema Ye., Grebelnykova V. The Optimization of the Video Data

- Compression Based on Wavelet Transformation. Proceedings of the Conference on Interactive Computer Aided Learning ICL2006. Villach, Austria, 2006. P. 1–6.
68. Sulema Ye. Multimedia and Computer Technologies in e-Learning. Proceedings of the International Conference “Informational Technologies for Rural Development”. Jelgava, Latvia, 2006. P. 1–4.
69. Sulema Ye. Multimedia in Computer Networks: Technologies and Applications. Proceedings of the 10<sup>th</sup> International Netties Conference. St. Pölten, Austria, 2005. P. 1–5.
70. Sulema Ye. Using Video and Static Panoramic Images in E-Learning. Proceedings of the International Conference ICL2005. Villach, Austria, 2005. P. 1–4.
71. Sulema Ye., Sherman Eu. Secure Data Transfer in Networks. Proceedings of the Symposium on Remote Engineering Virtual Instrumentation REV 2004. Austria, 2004. P. 1–8.
72. Sulema Ye., Bhattacharya A., Murray N. Multimedia Data Representation Based on Multi-Image Concept. Proceedings of the International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2019). Thessaloniki, Greece, 2019. P. 1–12.

***Всього за тематикою дисертації:***

За тематикою дисертаційної роботи опубліковано 72 наукові роботи. Основні наукові результати висвітлені в 26 наукових публікаціях.

Зокрема, опубліковано 1 монографію; 19 статей в наукових фахових журналах України, в тому числі 1 статтю у журналі категорії «А»; 6 наукових статей у закордонних виданнях, проіндексованих у базі даних Scopus, з яких 4 статті в журналах 3-го квартиля (Q3) та 1 стаття в журналі 4-го квартиля (Q4).

Крім того, результати роботи відображено у 7 статтях в інших наукових журналах України та за кордоном, а також у 39 публікаціях у матеріалах науково-технічних конференцій, з яких 10 видань проіндексовано у базі даних Scopus.

***Total for the dissertation theses:***

On the topic of the dissertation, 72 scientific publications have been published.

The main scientific results are shown in 26 scientific publications.

In particular, the following scientific publications have been published:

1 monograph; 19 articles in the professional scientific journals of Ukraine, including 1 article in a journal of the category «A»; 6 articles in professional foreign journals indexed in the reference base Scopus, where 4 articles are in the journals of 3-rd quartile (Q3) and 1 article is in the journals of 4-th quartile (Q4).

Besides, the results are presented in 7 articles in other scientific journals of Ukraine and abroad as well as in 39 proceedings of scientific conferences, including 10 proceedings indexed in the reference base Scopus.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ .....	25
ВСТУП .....	27
РОЗДІЛ 1. СУЧАСНИЙ СТАН ВИРІШЕННЯ ПРОБЛЕМИ ПОДАННЯ ТА ОБРОБКИ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВИХ ДВІЙНИКІВ ДОСЛІДЖУВАНИХ ОБ'ЄКТІВ .....	36
1.1. Концепція цифрового двійника досліджуваного об'єкта.....	36
1.2. Аналіз методів, моделей та програмних засобів обробки мультимодальних даних .....	42
1.2.1. Технологія цифрових двійників .....	42
1.2.2. Технологія мультимедіа .....	51
1.3. Аналіз парадигм програмування для розроблення програмного забезпечення систем обробки темпоральних мультимодальних даних .....	56
1.4. Висновки до розділу 1 .....	62
РОЗДІЛ 2. ТЕОРЕТИЧНІ ОСНОВИ ПОДАННЯ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВОГО ДВІЙНИКА ДЛЯ ОПИСУ ДОСЛІДЖУВАНОВОГО ОБ'ЄКТА .....	65
2.1. Алгебраїчна система агрегатів як інструмент формального опису досліджуваного об'єкта .....	66
2.2. Операції в алгебраїчній системі агрегатів .....	71
2.2.1. Логічні операції над агрегатами .....	71
2.2.2. Операції впорядкування .....	79
2.2.3. Арифметичні операції .....	88
2.3. Відношення в алгебраїчній системі агрегатів .....	88
2.3.1. Відношення між кортежами .....	89
2.3.2. Відношення над дискретними інтервалами .....	93
2.3.3. Відношення між агрегатами .....	102

2.4.	Нечіткі значення в алгебраїчній системі агрегатів .....	105
2.5.	Порівняльний аналіз алгебраїчної системи агрегатів .....	113
2.6.	Висновки до розділу 2 .....	115
РОЗДІЛ 3. МОДЕЛІ ПОДАННЯ ТА МЕТОДИ ОБРОБКИ ТЕМПО-		
РАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВОГО ДВІЙНИКА		
ДОСЛІДЖУВАНОВОГО ОБ'ЄКТА .....		117
3.1.	Концепція мультиобrazу для формальної специфікації цифрового	
	двійника .....	117
3.2.	Метод створення мультиобrazу досліджуваного об'єкта .....	124
3.3.	Метод синхронізації мультиобrazів .....	130
3.3.1.	Базові правила синхронізації .....	132
3.3.2.	Правила нечіткої синхронізації .....	163
3.3.3.	Процедура синхронізації мультиобrazів .....	171
3.4.	Моделі цифрового двійника .....	177
3.4.1.	Муксельна модель .....	177
3.4.2.	Часова зв'язна модель .....	193
3.4.3.	Багаторівнева онтологічна модель.....	198
3.5.	Метод адаптивного ущільнення графічних даних цифрового	
	двійника .....	201
3.6.	Метод конфіденційного зберігання даних цифрового двійника .	213
3.6.1.	Процедура шифрування даних з використанням палітри	
	значень .....	215
3.6.2.	Процедура стеганографічного приховування даних з	
	використанням булевих функцій .....	218
3.7.	Висновки до розділу 3 .....	220
РОЗДІЛ 4. ПАРАДИГМА ПРОГРАМУВАННЯ МУЛЬТИОБРАЗІВ ДЛЯ		
ОБРОБКИ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ		
ЦИФРОВИХ ДВІЙНИКІВ .....		223
4.1.	Парадигма програмування мультиобrazів .....	224

4.2.	Моделі обчислень для реалізації парадигми програмування мультимедіа.....	225
4.2.1.	Базова модель обчислень .....	226
4.2.2.	Модель обчислень для технології цифрових двійників .....	228
4.3.	Застосування парадигми програмування мультимедіа .....	230
4.4.	Спеціалізована мова програмування ASAMPL для обробки темпоральних мультимедіальних даних .....	233
4.4.1.	Синтаксис мови ASAMPL .....	235
4.4.2.	Семантика мови ASAMPL .....	249
4.4.3.	Прагматика мови ASAMPL .....	259
4.4.4.	Порівняльний аналіз мови ASAMPL .....	263
4.5.	Висновки до розділу 4.....	269
РОЗДІЛ 5. ПРОГРАМНІ СИСТЕМИ ОБРОБКИ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВИХ ДВІЙНИКІВ ДОСЛІДЖУВАНИХ ОБ’ЄКТІВ .....		271
5.1.	Архітектура програмної системи обробки темпоральних мультимедіальних даних цифрових двійників.....	271
5.2.	Програмне забезпечення медичних діагностичних систем на основі технології цифрових двійників .....	277
5.3.	Програмні системи дистанційного навчання на основі технології цифрових двійників та технології мультимедіа .....	285
5.3.1.	Архітектура онлайн лабораторії .....	285
5.3.2.	Архітектура імерсійного середовища .....	289
5.4.	Висновки до розділу 5.....	293
ВИСНОВКИ .....		296
СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ .....		299
ДОДАТОК А. Акти використання результатів дисертації.....		333
ДОДАТОК Б. Перелік наукових публікацій за темою дисертації та відомості про апробацію результатів.....		336



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

AutomationML	– Automation Markup Language (мова автоматичної розмітки)
B2MML	– Business-to-Manufacturing Markup Language (мова розмітки зв'язку між бізнесом та виробництвом)
DTDL	– Digital Twins Definition Language (мова визначення цифрових двійників)
EBNF	– Extended Backus-Naur Form (розширена форма Бекуса-Наура)
EOF	– End of File (маркер кінця файлу даних)
FMO	– Fuzzy Max Order (метод визначення нечіткого порядку)
JSON	– JavaScript Object Notation (текстовий формат файлу для обміну даними)
JSON-LD	– JSON for Linked Data (формат JSON для зв'язних даних)
LZW	– Lempel-Ziv-Welch (метод Лемпеля-Зіва-Велча)
LSB	– Less Significant Bit (найменш значущі біти)
MSB	– Most Significant Bit (найбільш значущі біти)
Multimedia	– Multiple Media (мультимедіа – технологія, що передбачає використання у комп'ютерних системах візуальної (графічної, текстової) та акустичної інформації)
Mulsemmedia	– Multiple Sensorial Media (мультисенсорна технологія, що передбачає використання у комп'ютерних системах всіх видів інформації, що сприймаються органами чуття людини)
MVPL	– Microsoft Visual Programming Language (мова візуального програмування Майкрософта)
NCL	– Nested Context Language (вкладена контекстна мова – спеціалізована мова програмування для розроблення гіпертекстових застосунків)
OpenGL	– Open Graphics Library (відкрита графічна бібліотека)

PVO	– Possibilistic Variation Order (метод ймовірнісного встановлення нечіткого порядку)
RGB	– Red-Green-Blue (колірна модель)
RLE	– Run Length Encoding (метод групового кодування)
QoS	– Quality of Service (якість обслуговування)
QoE	– Quality of Experience (якість користувацького досвіду)
SMIL	– Synchronized Multimedia Integration Language (мова інтеграції синхронізованого мультимедійного контенту)
STL	– Stereolithography (формат файлу для подання моделей, що виготовлятимуться шляхом тривимірного друку)
XML	– Extensible Markup Language (розширювана мова розмітки)
XSD	– XML Schema language (мова схем XML)
AK	– алгебра кортежів
ACA	– алгебраїчна система агрегатів
БП	– базове правило синхронізації
НП	– правило нечіткої синхронізації
ТМ	– теорія множин
ТММ	– теорія мультимножин
ТЧР	– теорія часових рядів
УП	– універсальне правило синхронізації
ШС	– шаблон синхронізації

## ВСТУП

**Актуальність роботи.** Сучасний стан розвитку науки, техніки та суспільства характеризується кількома тенденціями, які демонструють взаємний вплив. Одна з цих тенденцій, яка пов'язана з розвитком інформаційного суспільства, полягає у подальшій діджиталізації людської діяльності. Інформаційні технології та комп'ютерна техніка використовуються практично в усіх сферах діяльності, з'являються нові галузі їх застосування, рівень впровадження інформаційно-комунікаційних технологій поглиблюється, всі більше функцій, які були прерогативою людини, покладаються на обчислювальну техніку. Інша тенденція, яка пов'язана з розвитком науки і техніки, полягає у появі нових технологій та технічних засобів, які роблять можливим людино-центрований підхід до розроблення програмно-апаратного забезпечення, більшу адаптацію людино-машинних інтерфейсів до потреб користувачів, підвищення рівня ергономічності програмних продуктів тощо. Все це створює принципово нові можливості для подання та обробки у комп'ютерних системах інформації про об'єкти реального світу, з якими людина взаємодіє у виробничій, освітній та повсякденній діяльності.

Водночас, упродовж останніх десятиліть людство стикається з новими викликами, які виникають через техногенні катастрофи, природні катаклізми та епідемії, як-то Чорнобильська катастрофа, аварії на АЕС Фукусіма-1 (Японія), заводи хімічних відходів м. Кельце (Польща), виверження вулкана Ейяф'ятлайокютль (Ісландія), пандемія COVID-2019 тощо. Ці та інші подібні приклади наочно продемонстрували, що в деяких ситуаціях виникає необхідність у дистанційному форматі участі людини у вирішенні виробничих задач. При цьому в багатьох випадках є потреба у наданні людині (оператору, керівнику, користувачу тощо) всебічної інформації, що сприймається органами чуття людини, для забезпечення

повноцінного заглиблення в ситуацію. Тому розроблення нового класу програмно-апаратних систем на основі технології цифрових двійників, технології мультимедіа, імерсійних технологій дозволить розширити можливості та підвищити ефективність людської діяльності в складних або нестандартних умовах.

Технологія цифрових двійників розроблена фахівцями Національного агентства з авіації та дослідження космічного простору США для інженерії та виробництва, але основна її ідея може бути адаптована для інших галузей людської діяльності. Оскільки цифровий двійник характеризується не лише технічними характеристиками або поведінковими даними, але й візуальною моделлю, то доцільно розвивати технологію цифрових двійників у взаємозв'язку з технологією мультимедіа, яка передбачає оперування не лише аудіовізуальними даними, але й даними інших модальностей, які передають інші типи інформації, що сприймається органами чуття людини.

Спільне застосування технології цифрових двійників та технології мультимедіа дозволить синергетично посилити можливості обох технологій та сприятиме розробленню програмно-апаратних систем нового класу для вирішення людиною широкого кола задач у дистанційному форматі у ситуаціях, пов'язаних з ризиком або обмеженням можливості безпосередньої участі людини.

Незважаючи на наявність достатньої кількості науково-технічних рішень на основі технології цифрових двійників, на сьогодні відсутній системний підхід до вирішення задачі створення цифрових двійників, який враховував би всі аспекти застосування цієї технології: від розроблення моделей та оброблення мультимодальних даних цифрового двійника до захисту інформації, яка характеризує досліджуваний об'єкт. Тому в роботі пропонується комплексний підхід до розроблення програмного забезпечення систем, що ґрунтуються на основі технології цифрових двійників.

### **Зв'язок роботи з науковими програмами, планами, темами.**

Дослідження, представлені у дисертації, проводились в рамках держбюджетних науково-дослідних робіт Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» «Математичні та програмні методи оброблення мультимодальних даних моніторингу медико-біологічних об'єктів для діагностики стану здоров'я пацієнтів» (номер державної реєстрації 0120U102134) та «Розроблення та дослідження методів оброблення, розпізнавання, захисту та зберігання медичних зображень в розподілених комп'ютерних системах» (номер державної реєстрації 0117U004267); ініціативної науково-дослідної роботи «Методи та засоби обробки, захисту та пошуку мультимедійних даних у комп'ютерних системах та мережах» (реєстраційний номер 0114U003948); міжнародного наукового проєкту «Future eHealth powered by 5G» (номер державної реєстрації 0118U100449); проєкту програми Horizon2020 «Approximation Methods for Molecular Modelling and Diagnosis Tools» (реєстраційний номер 645672); проєкту програми FP7 «EU-Ukrainian Mathematicians for Life Sciences» (реєстраційний номер 295164).

**Мета і завдання дослідження.** Метою дисертаційної роботи є підвищення ефективності обробки темпоральних мультимодальних даних у комп'ютерних системах.

### **Задачі дослідження.**

1. Дослідити можливість та доцільність спільного застосування технології цифрових двійників та технології мультимедіа для подання та обробки темпоральних мультимодальних даних досліджуваних об'єктів.

2. Розробити метод створення мультиобразу досліджуваного об'єкта для формальної специфікації моделей цифрового двійника.

3. Розробити метод синхронізації мультиобразів, в тому числі, із застосуванням нечіткої логіки для вирішення задач з нечітко визначеними

вхідними даними.

4. Розробити апарат алгебраїчної системи агрегатів для формального опису досліджуваного об'єкта.

5. Розробити моделі цифрового двійника для комплексного подання інформації про досліджуваний об'єкт на різних рівнях абстракції.

6. Розробити парадигму програмування та моделі обчислень для забезпечення ефективної обробки темпоральних мультимодальних даних та спрощення процесу розроблення програмного забезпечення систем обробки даних цифрових двійників досліджуваних об'єктів.

7. Розробити спеціалізовану мову програмування для обробки темпоральних мультимодальних даних цифрового двійника досліджуваного об'єкта.

8. Розробити архітектуру програмної системи для обробки темпоральних мультимодальних даних цифрових двійників та дослідити можливість її застосування для вирішення прикладних задач у галузі охорони здоров'я та у галузі освіти.

**Об'єкт дослідження** – процеси обробки темпоральних мультимодальних даних.

**Предмет дослідження** – методи обробки та моделі подання темпоральних мультимодальних даних у комп'ютерних системах.

**Методи дослідження.** Для розв'язання поставлених задач використовуються основні положення: теорії алгоритмів та теорії програмування – для розроблення парадигми програмування, моделей обчислень та мови програмування; нечіткої логіки – для подання нечітких значень у алгебраїчній системі агрегатів; інтервальної алгебри Аллена – для формалізації відношень дискретних інтервалів; методів комп'ютерної графіки, теорії інформації і кодування – для розроблення методу адаптивного ущільнення графічних даних; методів захисту даних – для розроблення методу подання мультимодальних конфіденційних даних

цифрових двійників; теорії програмних систем та методів системного аналізу – для розроблення архітектури програмних систем обробки темпоральних мультимодальних даних цифрових двійників.

### **Наукова новизна одержаних результатів.**

Дістали подальший розвиток основи побудови програмного забезпечення систем обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів.

1. *Уперше* для формальної специфікації моделей цифрового двійника розроблено метод створення мультиобразу досліджуваного об'єкта, який полягає в послідовному виконанні процедур формування структури даних, декомпозиції мультиобразу на набір часткових мультиобразів, підготовки темпоральних мультимодальних даних, об'єднанні, сортуванні та проріджуванні даних, що дає змогу розробляти універсальне програмне забезпечення систем обробки даних цифрових двійників досліджуваних об'єктів.

2. *Уперше* розроблено метод синхронізації мультиобразів, який ґрунтується на застосуванні інтервальних відношень, виконанні правил синхронізації, в тому числі нечіткої, а також процедур формування шаблону синхронізації, впорядкування мультиобразів, попарної синхронізації мультиобразів, що робить можливим виконання консолідованої обробки інформації, яка надходить від довільної кількості досліджуваних об'єктів.

3. *Уперше* для опису досліджуваного об'єкта розроблено алгебраїчну систему, що задається трьома множинами – множиною елементів (агрегатів), множиною операцій та множиною відношень, яка відрізняється від існуючих врахуванням порядку слідування елементів при виконанні операцій, введенням операцій впорядкування та спеціальних відношень (частотних, інтервальних) між елементами, що дає змогу спростити алгоритмічно-програмне забезпечення систем обробки темпоральних мультимодальних даних, в тому числі нечітко визначених.

4. *Уперше* для подання темпоральних мультимодальних даних цифрового двійника досліджуваного об'єкта розроблено моделі цифрового двійника – муксельну модель, часову зв'язну модель та багаторівневу онтологічну модель, що уможлиблює використання різних рівнів абстракції для отримання комплексного опису досліджуваного об'єкта.

5. *Уперше* розроблено парадигму програмування, визначальною рисою якої є використання нової сутності – мультиобразу досліджуваного об'єкта як основного компонента обчислювального процесу, що дозволяє спростити процес розроблення програмного забезпечення систем обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів.

6. *Уперше* розроблено мову програмування ASAMPL (Algebraic System of Aggregates, Multimodal Processing Language), яка реалізує новорозроблену парадигму програмування, і спеціалізовані команди якої оперують мультиобразами та їх компонентами, що дає змогу отримувати програмний код з поліпшеними метриками.

7. *Уперше* запропоновано архітектуру програмної системи для обробки темпоральних мультимодальних даних цифрових двійників, особливістю якої є використання дворівневої процедури синхронізації та агрегації даних, що дозволяє спростити процес розроблення прикладного програмного забезпечення систем обробки темпоральних мультимодальних даних. Запропонована архітектура може бути основою для розроблення програмного забезпечення медичних діагностичних систем, а також онлайн-лабораторій та імерсійних середовищ у галузі освіти.

**Практичне значення одержаних результатів.** Одержані результати можуть бути застосовані для створення нового класу програмного забезпечення на основі технології цифрових двійників, зокрема, для розроблення медичних експертних систем нового покоління з розширеними можливостями щодо обробки та аналізу даних медичної діагностики, для



розроблення систем дистанційного навчання на основі технології цифрових двійників та технології мультимедіа.

Теоретичні та практичні результати роботи впроваджено у Національній медичній академії післядипломної освіти імені П. Л. Шупіка (акт про впровадження від 12.12.2019 р.), ТОВ «Відео Інтернет Технології» (акт про впровадження від 18.12.2019).

**Особистий внесок здобувача.** Всі основні результати дисертаційного дослідження, які представлені до захисту, одержані автором особисто.

У працях, опублікованих у співавторстві, особистий внесок здобувача є наступним: [1, 12, 13, 72] – розроблено методи обробки темпоральних мультимодальних даних та апарат алгебраїчної системи агрегатів; [2, 4, 6] – розроблено парадигму програмування мультиобразів, моделі обчислень, синтаксис та семантику мови програмування ASAMPL; [5, 26] – розроблено моделі подання темпоральних мультимодальних даних; [7, 9, 10, 14, 15, 17, 22, 24, 27, 28, 30–32, 35, 36, 43, 44, 47, 49, 51, 53, 59–62, 64, 65, 67] – розроблено методику обробки мультимедійних та мультимедійних даних; [8, 16, 18–20, 23, 25, 29, 50, 54–57, 63, 71] – розроблено методику подання конфіденційних мультимодальних даних; [11, 33, 34, 37, 45, 46, 48, 52, 58, 66] – розроблено архітектуру програмного забезпечення цифрових двійників; [38, 42] – виконано аналіз можливості застосування технології мультимедіа.

Решта робіт написані без співавторів.

**Апробація результатів дисертації.** Основні положення та результати проведених у дисертаційній роботі досліджень доповідались та обговорювались на 31 міжнародній та всеукраїнській науково-технічній конференції: International Conference on Interactive Mobile Communication, Technologies and Learning IMCL2019, IMCL2017, IMCL2015 (Thessaloniki, Greece, 2019, 2017, 2015), International Conference on Interactive Collaborative

and Blended Learning ICBL2019 (Santiago de Cuba, Cuba, 2019), 24th International Conference «European Wireless» EW2018 (Catania, Italy, 2018), International Conference on Advanced Computer Information Technologies ACIT 2018 (Ceske Budejovice, Czechia, 2018), International Workshop on Multimedia Information and Communication Technologies Redžúr 2018, Redžúr 2017, Redžúr 2016 (Bratislava, Slovakia, 2018, 2017, 2016), Symposium on Remote Engineering Virtual Instrumentation REV2018, REV2004 (Duesseldorf, Germany, 2018, Villach, Austria, 2004), International Symposium ELMAR-2017, ELMAR-2016 (Zadar, Croatia, 2017, 2016), 23rd IEEE International Conference on Systems, Signals and Image Processing IWSSIP2016 (Bratislava, Slovakia, 2016), 3rd International Conference Health Technology Management HTM-2016 (Chisinau, Moldova, 2016), «AMMODIT and final EUMLS Workshop» (Hasenwinkel, Germany, 2016), International conference «Information Technology. Problems and Solutions» (Ufa, Russia, 2013), IEEE International Conference on Automation, Quality and Testing, Robotics AQTR 2010 (Cluj-Napoca, Romania, 2010), IEEE International Conference East-West Design & Test International Symposium EWDTTS'09 (Moscow, Russia, 2009), International Conference on Interactive Computer Aided Learning ICL2007, ICL2006, ICL2005 (Villach, Austria, 2007, 2006, 2005), International Conference Information Technologies for Rural Development (Jelgava, Latvia, 2006), 10th International Netties Conference (St. Pölten, Austria, 2005), 6-й міжнародній науково-практичній конференції з сучасних проблем кодування, захисту й ущільнення інформації (Вінниця, Україна, 2017), міжнародній конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2014, АПКТ-2012, АПКТ-2011 (Хмельницький, Україна, 2014, 2012, 2011), міжнародній науково-технічній конференції «Системний аналіз та інформаційні технології» SAIT 2011 (Київ, Україна, 2011), міжнародній науково-технічній конференції «Современные информационные и электронные технологии» (Одесса, Україна, 2009), XVI міжнародній

конференції з автоматичного управління «Автоматика-2009» (Чернівці, Україна, 2009).

**Публікації.** Результати досліджень за тематикою дисертації опубліковано у 72 наукових працях. Основні наукові результати опубліковано у 26 наукових працях, з яких 1 монографія; 25 наукових статей (в тому числі 6 наукових статей у закордонних виданнях, проіндексованих у наукометричній базі Scopus, з яких 4 статті в журналах 3-го квартіля (Q3) та 1 стаття в журналі 4-го квартіля (Q4); 19 статей у наукових фахових виданнях України, в тому числі 1 стаття у журналі категорії А). Результати роботи також представлено у 7 статтях (в тому числі 1 стаття у іноземному виданні та 6 статей опубліковано у наукових виданнях України) та у 39 матеріалах наукових та науково-технічних конференцій (в тому числі у 10 матеріалах закордонних наукових конференцій, що реферуються наукометричними базами Scopus та/або Web of Science, і у 29 матеріалах інших наукових конференцій в Україні та за кордоном).

**Структура дисертації.** Дисертація складається зі вступу, п'яти розділів, висновків, списку використаних джерел, додатків. Робота містить 277 сторінок основного тексту, 9 таблиць, 52 рисунки, 367 найменувань використаних джерел, 2 додатки. Загальний обсяг дисертації становить 343 сторінки.

## **РОЗДІЛ 1. СУЧАСНИЙ СТАН ВИРІШЕННЯ ПРОБЛЕМИ ПОДАННЯ ТА ОБРОБКИ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВИХ ДВІЙНИКІВ ДОСЛІДЖУВАНИХ ОБ'ЄКТІВ**

Задача дослідження фізичного об'єкта включає в себе декілька важливих підзадач, пов'язаних з отриманням даних про об'єкт дослідження, передньою обробкою цих даних, поданням даних у вигляді деякої сукупності (структури даних) та подальшою обробкою, логіка якої визначається дослідником залежно від мети дослідження. Спорідненими задачами, які можуть вирішуватись у рамках виконання дослідження об'єкта, є моделювання його стану, аналіз даних, які відображають характеристики об'єкта та визначають його поведінку, а також відображають зовнішні риси об'єкта в цілому та його складових частин. Для виконання цих та інших споріднених задач доцільно представляти отриманні від фізичного об'єкта дані у вигляді комплексної моделі, яку називають *цифровим двійником* (digital twin) [73]. Для виявлення проблем, що пов'язані з поданням та обробкою мультимодальних даних цифрових двійників досліджуваних об'єктів, проаналізуємо концепцію цифрового двійника, а також існуючі методи, моделі та програмні засоби, що використовується для вирішення науково-технічних задач, пов'язаних з дослідженням фізичного об'єкта.

### **1.1. Концепція цифрового двійника досліджуваного об'єкта**

Технологія цифрових двійників реалізує відносно новий підхід [73–79] до подання у цифровій формі фізичного об'єкта чи процесу, його минулого, поточного і майбутнього стану, поведінки, характеристик, зовнішнього вигляду тощо. Відповідно до визначення, сформульованого авторами концепції цифрового двійника, Майклом Грівзом (Michael

Grieves) та Джоном Вікерсом (John Vickers), у роботі [73], *цифровий двійник* – це сукупність віртуальних інформаційних конструкцій (virtual information constructs), яка повністю описує фізичний об’єкт – від мікрорівня (рівень окремого елементу) до макрорівня (загальний вигляд, геометричне подання, загальні властивості об’єкта в цілому). Цифровий двійник являє собою модель фізичного об’єкта або процесу (*фізичного двійника*), яка якнайповніше відображає його характеристики у динаміці протягом певного періоду часу. Концепція цифрового двійника передбачає подання, обробку, маніпулювання усіма даними, що характеризують фізичного двійника, та отримання певних компонент цих даних по мірі виникнення потреби у них при вирішенні задач аналізу та прогнозування стану фізичного двійника, оптимізації процедур керування фізичним двійником тощо.

Термін «digital twin», як такий, вперше згадувався у звіті Національного управління з аеронавтики і дослідження космічного простору США [80], де визначено, що цифровий двійник є інтегрованою мультифізичною (multiphysics), мультимасштабованою (multiscale), ймовірнісною симуляцією об’єкта дослідження (наприклад, літального апарата), яка використовує найкращі наявні моделі, що постійно вдосконалюються, дані, що безперервно надходять з давачів реального об’єкта та накопичуються, щоб відобразити життєвий цикл відповідного досліджуваного об’єкта – фізичного двійника. Цифровий двійник має забезпечувати реалістичність відображення фізичного двійника.

Аналіз даних цифрового двійника дозволяє виявити аномалії поведінки складових фізичного двійника до виникнення аварійної ситуації у фізичному двійнику та, таким чином, вчасно запобігти виникненню надзвичайній ситуації. Наприклад, цифровий двійник літального апарату [80–83] інтегрує дані, що отримуються з давачів вбудованої системи, що знаходиться на борту фізичного двійника, містить дані про історію

обслуговування досліджуваного об'єкта, історію його аварійних ситуацій тощо. Таким чином, цифровий двійник є *індивідуалізованою* реалістичною віртуальною моделлю певного досліджуваного об'єкта – фізичного двійника.

Цифровий двійник складається з *візуальної моделі* досліджуваного об'єкта та *поведінкової моделі*, що реалізовані на основі відповідних математичних моделей і моделей подання даних та забезпечують синхронізацію між віртуальною і реальною системою на рівні даних, що надходять із давачів, встановлених для постійного моніторингу досліджуваного об'єкта.

З точки зору практичної реалізації цифровий двійник являє собою складну програмну систему, яка включає сховище даних та програмні модулі, що забезпечують отримання, обробку та візуалізацію даних.

За призначенням розрізняють [73, 79] цифрові двійники двох основних типів: цифровий двійник-прототип (*digital twin prototype*) та цифровий двійник-екземпляр (*digital twin instance*).

*Цифровий двійник-прототип* [73, 79] створюється для фізичного об'єкта, який ще не існує та буде створений після дослідження цифрового двійника. Він містить набори даних, необхідні для опису та створення фізичного двійника. Ці набори даних включають вимоги до фізичного двійника, тривимірну модель фізичного об'єкта, специфікацію матеріалів та компонентів, з яких він буде вироблений тощо.

*Цифровий двійник-екземпляр* [73, 79] створюється для конкретного існуючого фізичного об'єкта, з яким цей цифровий двійник залишається пов'язаним протягом усього життєвого циклу або циклу дослідження фізичного двійника.

Окрім цифрового двійника-прототипу та цифрового двійника-екземпляру, розрізняють *цифровий двійник-агрегат* (*digital twin aggregate*) [73, 79], який отримується в результаті агрегації деякої сукупності

цифрових двійників-екземплярів та використовується для дослідження певного набору (класу) фізичних об'єктів, а не окремого досліджуваного об'єкта.

*Середовище цифрового двійника (digital twin environment)* [73] – це сукупність програмного та програмно-апаратного забезпечення процесу дослідження одного або декількох фізичних об'єктів. Середовище цифрового двійника має дві основні функції.

1. **Передбачувальна функція.** Цифровий двійник використовується для прогнозування майбутньої поведінки фізичного двійника. У випадку застосування цифрового двійника-прототипу прогнозування дозволяє проаналізувати різні варіанти реалізації фізичного двійника ще до його фізичного створення. У випадку застосування цифрового двійника-екземпляру прогнозування має на меті передбачення можливих критичних ситуацій для фізичного двійника та моделювання розвитку стану фізичного двійника залежно від варіантів впливу (взаємодії, керування) на цей фізичний об'єкт або процес.
2. **Інформаційна функція.** Використання цифрового двійника дозволяє отримати актуальну інформацію про досліджуваний об'єкт в цілому або про його конкретні характеристики. Ця функція є важливою для вирішення широкого кола дослідницьких задач.

За рівнем інтеграції цифрового та фізичного двійників розрізняють [84] цифрову модель (digital model), цифрову тінь (digital shadow) та, власно, цифровий двійник (рис. 1.1).

*Цифрова модель* (рис. 1.1a) – це цифрове подання існуючого або запланованого фізичного об'єкта, яке не використовує жодної форми автоматизованого обміну даними між фізичним об'єктом та цифровим об'єктом. Цифрова модель передбачає використання імітаційних моделей, математичних моделей або інших видів моделювання фізичного об'єкта, коли автоматична інтеграція даних відсутня. Цифрові дані існуючих

фізичних систем можуть використовуватися для розроблення таких моделей, але обмін даними здійснюється вручну. Зміна стану фізичного об'єкта не має прямого впливу на цифровий об'єкт і навпаки.

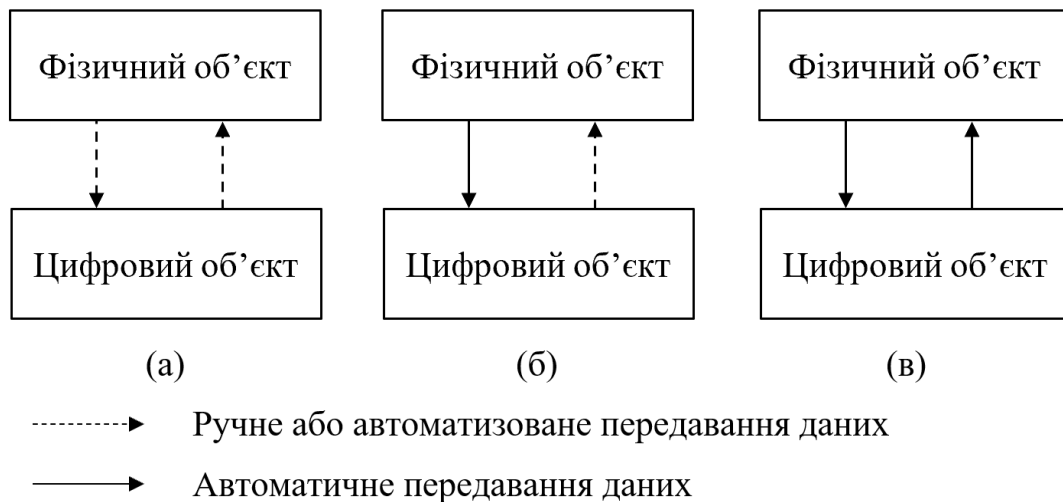


Рис. 1.1. Інтеграція фізичного та цифрового об'єктів: (а) цифрова модель, (б) цифрова тінь, (в) цифровий двійник

*Цифрова тінь* (рис. 1.1б) – це цифрове подання фізичного об'єкта, при якому передбачено автоматизований односторонній потік даних між існуючим фізичним об'єктом та цифровим об'єктом. Зміна стану фізичного об'єкта призводить до зміни стану цифрового об'єкта, але не навпаки.

*Цифровий двійник* (рис. 1.1в) – це цифрове подання фізичного об'єкта, при якому потоки даних між існуючим фізичним об'єктом і цифровим об'єктом наявні в обох напрямках. Такий цифровий об'єкт може розглядатись як контрольний примірник фізичного об'єкта. Також можливе одночасне використання інших об'єктів, фізичних чи цифрових, які можуть викликати зміни стану в цифровому об'єкті. Зміна стану фізичного об'єкта безпосередньо призводить до зміни стану цифрового об'єкта і навпаки.

Для дослідження фізичного об'єкта доцільно використовувати цифровий двійник типу «цифрова тінь».



За рівнем складності візуальної моделі цифрового двійника розрізняють [85] цифрові двійники чотирьох рівнів: *підготовчий цифровий двійник* (pre-digital twin), *цифровий двійник*, *адаптивний цифровий двійник* (adaptive digital twin) та *інтелектуальний цифровий двійник* (intelligent digital twin) (табл. 1.1).

Таблиця 1.1 – Рівні складності цифрових двійників

Рівень цифрового двійника	Складність моделі	Зв'язок з фізичним двійником	Застосування машинного навчання
1-й рівень. Підготовчий цифровий двійник	Спрощена віртуальна модель	Не передбачено	Відсутнє
2-й рівень. Цифровий двійник	Віртуальна модель фізичного двійника	Пакетні дані	Відсутнє
3-й рівень. Адаптивний цифровий двійник	Віртуальна модель фізичного двійника з адаптивним користувацьким інтерфейсом	Дані у реальному часі	Частково
4-й рівень. Інтелектуальний цифровий двійник	Віртуальна модель фізичного двійника з адаптивним користувацьким інтерфейсом та застосуванням алгоритмів навчання з підкріпленням	Пакетні дані / дані у реальному часі	Наявне

Для дослідження фізичного об'єкта може використовуватись цифровий двійник другого, третього та четвертого рівнів.

Розглянемо існуючі методи, моделі та програмні засоби обробки мультимодальних даних цифрових двійників.

## **1.2. Аналіз методів, моделей та програмних засобів обробки мультимодальних даних**

Подальший розвиток концепції цифрового двійника можливий в кількох напрямках, зокрема, у напрямку застосування засобів штучного інтелекту та у напрямку переходу від графічного до мультимедійного та мультисемедійного подання досліджуваного об'єкта. В цьому дисертаційному дослідженні пропонується поєднання можливостей, що надають технологія цифрових двійників та технологія мультисемедіа. Проаналізуємо методи, моделі та програмні засоби обробки мультимодальних даних для кожної з цих технологій.

### **1.2.1. Технологія цифрових двійників**

Переважає більшість досліджень [73–114], які опубліковані за тематикою цифрових двійників, присвячена загальним питанням щодо технології цифрових двійників, галузі її застосування, перевагам, які надає ця технологія, та організаційним питанням щодо її впровадження. Таке спрямування публікацій пояснюється тим, що цей напрям досліджень виник порівняно нещодавно, тому наукові роботи на цьому етапі зосереджені на вирішенні більш глобальних питань, що стосуються концепції цифрових двійників як такої.

Тим не менш, деякі дослідники пропонують рішення більш конкретних, практично-орієнтованих наукових задач.

Так, у роботах [86–88] представлено базову архітектуру програмної системи для обробки даних цифрового двійника. Архітектура, запропонована у роботі [86], передбачає наявність наступних основних компонентів:

1. Програмний компонент, що реалізує інформаційну модель, яка надає абстрактну специфікацію технічних характеристик фізичного об'єкта.
2. Програмний компонент, що реалізує механізм зв'язку між цифровим та фізичним двійниками.
3. Програмний компонент, що реалізує процедури обробки, аналізу та пошуку мультимодальних даних для отримання актуальної інформації про фізичний двійник.
4. Сховище пар «ключ-значення» [125], що призначене для зберігання даних цифрового двійника.

Запропонована архітектура не передбачає компонента для візуалізації цифрового двійника. Результатом роботи такої програмної системи є лише аналітичні дані.

У роботі [87] представлено базову архітектуру цифрового двійника для хмарних кібер-фізичних систем. Одним з компонентів запропонованої архітектури є інтелектуальний контролер взаємодії, що дозволяє динамічне переконфігурування кібер-фізичної системи.

Автори роботи [88] пропонують узагальнену програмно-апаратну архітектуру цифрового двійника, яка складається з шістьох рівнів. Першим, найнижчим, рівнем є рівень фізичного об'єкта (фізичного двійника), на якому реєструються його характеристики за допомогою набору датчиків та відбувається взаємодія з фізичним двійником за допомогою актуаторів. Другий рівень – це рівень локальних контролерів, які забезпечують специфічну функціональність цифрового двійника. Третім рівнем є рівень локальної бази даних. Четвертий рівень є комунікаційним, який забезпечує взаємодію локального та хмарного сховищ даних. П'ятий рівень – це хмарне сховище даних. Шостий, останній, рівень є рівнем, на якому відбувається емуляція (моделювання поточної поведінки) та симуляція (моделювання майбутньої або можливої поведінки) фізичного двійника.

Абстрактну модель цифрового двійника представлено у роботі [89]. Автори зазначають, що ця модель повністю описує фізичний двійник на концептуальному рівні протягом усього його життєвого циклу. Проте, запропонована абстрактна модель не передбачає подання деталізованої інформації про фізичний об'єкт.

У роботі [90] запропоновано базову модель цифрового двійника, яка призначена для аналізу та моделювання виробничих процесів. Вона складається з чотирьох рівнів: рівня фізичного простору виробничого процесу, рівня комунікаційної системи, рівня цифрового двійника та рівня користувацького простору. Програмна система для реалізації цифрового двійника включає модуль керування, модуль симуляції, модуль передбачення і виявлення аномалій, а також модуль збереження даних (хмарне сховище). Недоліком запропонованої моделі є неможливість її використання для інших галузей застосування цифрових двійників.

Збереження даних цифрового двійника може виконуватись у форматі AutomationML [91], що запропоновано у роботі [92]. Цей формат ґрунтується на форматі XML [126] та дозволяє зберігати дані моделей промислових об'єктів. Об'єкт подіється у вигляді ієрархії його складових. Отримана модель може бути використана для обміну даними між компонентами програмної платформи.

Іншим форматом, який може застосовуватись для збереження даних моделей промислових об'єктів є B2MML [127]. Цей формат також ґрунтується на форматі XML, а також стандарті IEC/ISO 62264 [128]. B2MML складається з набору схем XML, написаних за допомогою мови XSD [129].

Дослідження, представлені в [93, 94], присвячені створенню візуальних моделей цифрових двійників. Автори демонструють рішення цієї задачі на прикладі створення цифрового двійника мікро-виробничого блока (micro manufacturing unit) [93] та цифрового двійника вертикального

фрезерного верстата (vertical milling machine) [94]. Для отримання візуальної моделі пропонується використовувати програмну платформу SolidWorks [130]. Створена модель [94] може бути збережена у форматі STL [131], що дозволяє виконувати візуалізацію не лише на екрані монітору, а також у вигляді виготовленої шляхом тривимірного друку пластикової моделі. Візуалізація програмним шляхом виконується за допомогою бібліотеки OpenGL [132]. Моніторинг характеристик фізичного двійника виконується за допомогою набору давачів, дані з яких надходять для зберігання у локальне сховище даних – базу даних PostgreSQL [133]. Віддалений доступ до даних відбувається через мережу Інтернет. Мультимодальний набір даних для цифрового двійника вертикального фрезерного верстата [94] включає дані про вібрацію, прискорення, а також електричні характеристики (сила току, споживання енергії). Обробка темпоральних мультимодальних даних у цьому дослідженні має на меті передбачення ступеню якості механічної обробки деталей за допомогою верстата. Для досягнення цієї мети у роботі [94] пропонується відповідна математична модель.

В роботі [95] пропонується візуалізація цифрового двійника засобами доповненої реальності, які використовуються у промислових системах візуалізації [96–101]. Така візуалізація передбачає накладання на зображення фізичного об'єкта допоміжних зображень та/або текстових даних, які пояснюють людині-оператору сутність роботи з цим об'єктом, надають довідкову інформацію, попередження про правила безпеки тощо. Запропонований у роботі підхід не передбачає використання для відображення цифрового двійника даних інших модальностей.

Автори робіт [102, 103] концептуально вирішують проблему синхронізації цифрового та фізичного двійників за допомогою вузлових точок. Зокрема, запропонований у роботі [103] метод дозволяє забезпечити зв'язок між різними моделями промислового об'єкта для коректної

візуалізації цифрового двійника. Цей метод не може бути застосований для цифрового подання об'єктів іншої природи, наприклад, медико-біологічних об'єктів.

У роботі [104] автори розглядають фактори ризику експлуатації механізмів, які діагностовано за допомогою цифрового двійника. Також наведено основні помилки при проведенні діагностики з використанням цифрових двійників.

Автор роботи [105] пропонує модель «розумного вагона», яка передбачає використання архітектури програмної системи, що включає підсистему рівня фізичних об'єктів, мережеву підсистему, підсистему хмарної інфраструктури та підсистему шлюзів. Обробка аналітичних даних, що надходять від давачів, є двоетапною. На першому етапі обробка даних виконується у реальному часі за допомогою мікропроцесорів. На другому етапі відбувається інтелектуальна обробка даних за допомогою алгоритмів штучного інтелекту та нечіткої логіки.

Аналіз цих та інших літературних джерел [73–114] дозволяє стверджувати про відсутність усталених теоретичних основ розроблення цифрових двійників, що пояснюється тим, що цей науковий напрям знаходиться на початку свого розвитку. Існуючі підходи не є універсальними, а мають прикладний характер та вирішують задачу створення цифрового двійника для конкретного застосування.

Розглянемо програмні продукти [115–124], що можуть бути застосовані для реалізації технології цифрових двійників.

Прикладом програмної платформи, що частково підтримує технологію цифрових двійників, є IBM® Watson™ IoT Platform [115]. Це програмне забезпечення дозволяє отримувати дані про об'єкт моніторингу від набору давачів. Підтримуються декілька форматів даних, в тому числі JSON [134], текстовий формат та формат для збереження геопросторових даних. У програмній платформі передбачено три варіанти збереження

даних. Дані, що отримуються від давачів у реальному часі, зберігаються у NoSQL базі даних [135]. Дані, які потребують аналізу, зберігаються у хмарному сховищі, організованому за технологією Data Lake [136, 137]. Дані, що потребують зберігання тривалий час, зберігаються у хмарному сховищі об'єктів [138]. Водночас, у архітектурі цієї програмної платформи не передбачено компонент для формування та відображення цифрового двійника.

Програмне забезпечення Seebo Digital Twin [116] розроблене для підтримки застосування технології цифрових двійників у галузі виробництва. Воно призначене для моделювання роботи технологічної ліній, зокрема, дозволяє створювати графічні прототиби промислових продуктів та виконувати симуляцію їх виробництва для виявлення слабких місць у технологічному процесі. Проте, дане програмне забезпечення має обмежені можливості щодо зв'язку з фізичним двійником. Крім того, воно призначене для використання лише у галузі виробництва.

Створення візуальних симуляційних моделей цифрових двійників [117, 118] для промислових застосунків може виконуватись за допомогою спеціалізованого програмного забезпечення компанії Simio [119]. Для створення візуальної моделі може бути використана стандартна бібліотека об'єктів. Для симуляції цифрового двійника використовується редактор, який дозволяє призначати операції, для яких виконуватиметься симуляція, час виконання симуляції, зв'язок з іншими віртуальними об'єктами тощо. Для опису властивостей фізичного двійника у програмній системі визначено близько 50 типів даних, включаючи типи даних для опису елементів (матеріал, стан тощо) та типи даних для опису об'єктів (сутність, ресурс, вузол тощо). Галуззю застосування цього програмного продукту є промисловість.

Найбільш розвинуті можливості для створення цифрових двійників надає платформа Microsoft Azure Digital Twins [120]. Для опису моделей цифрових двійників використовується мова DTDL [139], яка ґрунтується на використанні формату JSON [134], а саме формату JSON-LD [140].

Для визначення поведінки цифрових двійників у мові DTDL визначено п'ять класів метамоделі: Telemetry, Property, Command, Relationship та Component. Клас Telemetry описує дані, що генеруються давачами. Клас Property описує стан цифрового двійника та його синхронізацію з іншими компонентами розподіленої системи. Клас Command описує функцію або операцію, що може бути виконана цифровим двійником. Клас Relationship описує зв'язок цифрового двійника з іншими цифровими двійниками (зв'язок між інтерфейсами). Клас Component описує вміст, який є частиною інтерфейсу.

Приклад програмного коду мовою DTDL, який визначає об'єкт «Телефон», наведено у лістингу 1.1.

Лістинг 1.1 – Приклад програмного коду мовою DTDL [139]

```
{  "@id": "dtmi:com:example:Phone;2",
  "@type": "Interface",
  "displayName": "Phone",
  "contents": [
    {
      "@type": "Component",
      "name": "frontCamera",
      "schema": "dtmi:com:example:Camera;3"
    },
    {
      "@type": "Component",
      "name": "backCamera",
      "schema": "dtmi:com:example:Camera;3"
    },
    {
      "@type": "Component",
      "name": "deviceInfo",
      "schema":
"dtmi:azure:deviceManagement:DeviceInformation;2"
    }
  ],
  "@context": "dtmi:dtdl:context;2" }
```



Для спрощення аналізу даних при створенні цифрового двійника можуть використовуватись анотації поведінки семантичного типу. Прикладами семантичних типів, що визначені у мові DTDL, які можна застосувати для анотування значень телеметрії та властивостей, є Acceleration (прискорення), Density (густина), Force (сила) тощо.

Базовими операціями над цифровими двійниками є створення, модифікація та вилучення. Для виконання цих операцій можуть бути використані прикладні програмні інтерфейси DigitalTwins API двох рівнів: управлінського рівня та рівня даних [141]. Прикладні програмні інтерфейси управлінського рівня використовуються для управління екземпляром Azure Digital Twins в цілому, а саме, для його створення або вилучення. Прикладні програмні інтерфейси рівня даних використовуються для управління елементами в екземплярі Azure Digital Twins та включають такі категорії: DigitalTwinsModels, DigitalTwins, Query та EventRoutes. Інтерфейси категорії DigitalTwinsModels призначені для управління моделями екземпляра Azure Digital Twins. Інтерфейси категорії DigitalTwins дозволяють створювати, модифікувати та вилучати цифрові двійники у екземплярі Azure Digital Twins. Інтерфейси категорії Query призначені для пошуку наборів цифрових двійників. Категорія EventRoutes містить інтерфейси для маршрутизації даних.

Для побудови цифрового двійника складного об'єкта (наприклад, конвеєрної лінії на виробництві) цифрові двійники окремих його компонентів можуть бути об'єднані за допомогою графової моделі. Для роботи з графовим поданням цифрового двійника призначено програмний продукт Azure Digital Twins Explorer [142], який дозволяє завантажувати моделі, задавати їх параметри та встановлювати зв'язки між компонентами складеного цифрового двійника.

Дані цифрового двійника зберігаються у сховищі Azure Data Lake [143]. Аналіз даних виконується за допомогою засобів Azure Synapse Analytics [144].

Призначенням платформи Azure Digital Twins є моделювання з метою оптимізації бізнес-процесів для великих об'єктів та складних систем (фабрики, ферми, стадіони, залізничні мережі, енергетичні системи тощо).

Іншими прикладами програмних платформ, що призначені для реалізації технології цифрових двійників або частково її підтримують, є Digital Twin Machining [121], CosmoTech Enterprise Digital Twin [122], SAP Digital Manufacturing Cloud [123] та продукти Autodesk [124]. Галуззю застосування цих програмних систем також є виробництво та організація бізнес-процесів, вони не призначені для застосування у інших сферах діяльності, наприклад, у медицині та освіті.

Таким чином, аналіз методів та програмних засобів, що можуть бути застосовані для подання та обробки мультимодальних даних цифрових двійників, дозволяє зробити такі висновки про недоліки існуючих методів, моделей та програмних систем обробки даних цифрових двійників:

1. Наявні програмні системи спрямовані на вирішення інженерних та організаційних задач у галузі виробництва та не призначені для застосування у інших галузях, зокрема, у медичній галузі та галузі дистанційного навчання.
2. Рівень реалістичності візуальних моделей, що створюються за допомогою існуючих програмних систем, у переважній більшості випадків є недостатнім. Візуальна модель цифрового двійника обмежується лише графічним поданням фізичного двійника, причому звичайною практикою є двовимірна візуалізація. Інші види мультимодальних даних у візуальних моделях не застосовуються.
3. Відсутній загальний (універсальний, системний) підхід до подання і обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів та розроблення програмного забезпечення систем, які ґрунтуються на концепції цифрового двійника, для широкого кола застосувань. Потенціал концепції цифрового двійника не

використовується у повній мірі для вирішення задач дослідження довільного об'єкта (процесу, суб'єкта).

Оскільки однією з проблем візуалізації цифрових двійників є недостатній рівень реалістичності відображення фізичного двійника, розглянемо шляхи вирішення цієї проблеми за допомогою технології мультимедіа.

### 1.2.2. Технологія мультимедіа

Як було зазначено вище, у програмному забезпеченні систем на основі технології цифрових двійників для відображення фізичного об'єкта використовується лише візуальна інформація. В ширшому контексті, для подання інформації про об'єкти реального світу у цифровому вигляді використовуються дві модальності – візуальна та акустична інформація, що позначається терміном *multimedia* (multimedia). Проте фізичний об'єкт може бути описаний набагато інформативніше. Так, з позицій антропологічного підходу досліджуваний об'єкт може бути поданий у цифровому вигляді за допомогою мультимодальної інформації, що сприймається людиною через множинні відчуття, які поділяються на дві основні групи:

- дистантні відчуття (зір, слух, нюх, відчуття температури тощо),
- контактні відчуття (смак, дотик, відчуття ваги, вібрації тощо).

Технологія *mulsemmedia* (mulsemmedia) [145–186], а також *імерсійні технології* (immersive technologies) [187–190] реалізують людиноцентрований підхід, що передбачає реєстрацію, подання, оброблення та відтворення мультимодальної інформації про фізичні об'єкти, яка сприймається людиною через органи чуття. Термін «mulsemmedia» уперше був запропонований Георгітом Гінеєм (Gheorghita Ghinea) у роботі [145].

Фізичні об'єкти, процеси та явища у більшості сприймаються людиною через множинні відчуття, що і лежить в основі їх цифрового подання за допомогою технології мультимедіа. Наприклад, серцебиття пацієнта може бути описане як мультимодальний сигнал, який включає в себе акустичну компоненту (серцебиття може бути почуте), тактильну компоненту (серцебиття може бути відчуте тактильно, що використовується при визначенні пульсу), а також воно може бути описане за допомогою візуального сигналу, який отримується, наприклад, в процесі ехокардіографічного дослідження, та електричного сигналу, який отримується в процесі електрокардіографічного дослідження. Поєднання цих мультимодальних даних дозволяє надати найповніший опис досліджуваного об'єкта – серця пацієнта. Таким чином, мультимодальна інформація може бути використана для комплексного подання властивостей фізичного об'єкта з метою отримання та подальшого використання його цифрового двійника.

За рахунок спільного застосування технології мультимедіа та технології цифрових двійників певні прикладні задачі можуть вирішуватись на якісно новому рівні. Наприклад, у задачах, пов'язаних з дистанційним спостереженням, застосування технології мультимедіа дозволить отримати більш повний опис фізичного двійника (пасивне спостереження), а додавання тактильного зв'язку надасть можливість взаємодії з об'єктом з метою отримання більш повної інформації про нього (активне спостереження). У галузі дистанційного навчання спільне застосування технології мультимедіа та технології цифрових двійників дозволить створювати навчальні засоби нового покоління – цифрові двійники з мультимедійним описом фізичних об'єктів. У задачах,

пов'язаних з пошуком інформації, створення баз мультимодальних даних та пошукових механізмів до них дозволить забезпечити доступ до інформації на якісно новому рівні. У галузі медицини запровадження нового класу програмних систем на основі технології мультимедіа та технології цифрових двійників дозволить надавати якісну спеціалізовану консультативну медичну допомогу пацієнтам у форматі телемедичних послуг. Врешті, поєднання технології мультимедіа та технології цифрових двійників сприятиме розробленню нового класу людино-машинних інтерфейсів для переходу на якісно новий рівень взаємодії між людиною та комп'ютерною системою, що дозволить створювати програмні системи для вирішення виробничих задач у складних або нестандартних умовах при дистанційному форматі участі людини.

Апаратне забезпечення мультимедійних систем включає пристрої для запису та відтворення тривимірної візуальної інформації [191–201], пристрої для запису та відтворення просторової акустичної інформації [152, 202–207], пристрої для тактильної взаємодії [154–156, 208–212], пристрої для визначення та відтворення запаху [152–154, 213–217], пристрої для симуляції смакових відчуттів [154, 157–162, 218], пристрої для створення інших сенсорних ефектів (зміна температури, вологості, відтворення ефекту вітру, вібрації, руху тощо) [189, 190]. Існуюче апаратне забезпечення мультимедіа дозволяє реалізувати візуальну модель цифрового двійника на достатньо високому технічному рівні. Згідно з класифікацією рівнів технологічної готовності [219] цей рівень може бути оцінений як TRL 7.

Більшість наукових досліджень в галузі мультимедіа стосуються вирішення таких складних задач, як синхронізація мультимодальних даних [148–152], організація людино-машинної взаємодії та створення нових користувацьких інтерфейсів на основі застосування мультимедійних даних

[153–162], розроблення мультимедійних застосунків [163–176, 184–186], підвищення рівня QoS та QoE для мультимедійних систем [177–183] тощо.

Дослідження [145, 148–152], які присвячені вирішенню проблеми синхронізації мультимодальних даних, є дотичними до наукової задачі розроблення методів синхронізації темпоральних мультимодальних даних цифрових двійників. Зокрема, у роботі [145] представлено результати дослідження, що спрямоване на уточнення часових меж, у яких мультимедійні компоненти можуть бути якісно синхронізовані, що є важливим при розробленні візуальних моделей цифрових двійників.

Автори робіт [148–150] пропонують способи синхронізації сенсорних ефектів за допомогою різних варіантів анотування мультимедійних компонентів. У роботі [151] проаналізований вплив некоректної синхронізації сенсорних ефектів на сприйняття мультимедійного матеріалу користувачем.

У роботі [152] автори пропонують темпоральні моделі для зв'язаних мультимедійних документів (моделі на основі точок, подій або інтервалів) та базові способи синхронізації даних, пропонують рішення численних прикладних задач, а також аналізують можливості застосування спеціалізованих мов програмування SMIL [220] та NCL [221] для синхронізації мультимедійних елементів. Мови програмування SMIL та NCL ґрунтуються на XML [126]. Мова програмування SMIL призначена для розроблення веб-орієнтованих мультимедійних застосунків та дозволяє оперувати такими мультимедійними елементами, як текст, зображення, відео та аудіо. Мова програмування NCL призначена для декларативної специфікації програмних застосунків на основі концепції гіпермедіа [222].

Дослідження [171, 184–186], які пов'язані з розробленням мультимедійних застосунків, є дотичними до наукової задачі удосконалення технології розроблення програмних систем обробки мультимодальних даних цифрових двійників, зокрема компонента, що

виконує відтворення візуальної моделі цифрового двійника. Так, у роботах [184, 185] представлено програмні засоби, які призначено для реалізації тактильної взаємодії між користувачем та віртуальними об'єктами. Редактор HFX Studio [184] ґрунтується на трьох перцептивних моделях: моделі тиску, вібрації та температури, які дозволяють просторову тактильну дію на організм користувача. Запропоновані моделі можуть також використовуватись для опису можливостей пристроїв з тактильним зв'язком.

Редактор H-Studio [185] призначений для додавання тактильних та рухових ефектів до аудіовізуального матеріалу. Для створення рухових ефектів запропоновано три методи: ручне створення за ключовими точками, запис траєкторії за допомогою пристрою зі зворотним тактильним зв'язком та імпорт даних про рух об'єкта.

Розподілену мультимедійну платформу PlaySEM представлено у роботі [186]. Автори пропонують тестовий сценарій для мультимедійного застосунку та виконують аналіз часової ефективності його роботи. Автори зазначають, що при віддаленому використанні актуаторів, які відтворюють сенсорні ефекти, через дротову мережу, затримка відтворення мультимедійних ефектів складає в середньому 27-67 мс, що є задовільним для більшості застосувань. Платформа PlaySEM є сумісною зі стандартом MPEG-V та стандартами ISO/IEC [155, 167, 223–226].

У роботі [171] представлено програмну платформу для створення мультимедійних застосунків. Вона включає три компоненти: редактор сенсорних ефектів SEVino [227], симулятор сенсорних ефектів SESim [228] та програвач сенсорних ефектів SEMP [229]. Програмна система ґрунтується на стандарті MPEG-V [224]. Компоненти платформи розроблено мовою Java (SEVino, SESim) та C++ (SEMP).

Аналіз програмного забезпечення, що підтримує технологію мультимедіа, продемонстрував:

- відсутність спеціалізованих мов програмування, що призначені для розроблення програмного забезпечення систем обробки мультимедійних даних;
- орієнтованість процесу розроблення мультимедійних застосунків на використання редакторів сенсорних ефектів, що додаються до аудіовізуального матеріалу шляхом анотування.

В разі створення мультимедійного програмного забезпечення його розроблення переважно виконується згідно з парадигмою об'єктно-орієнтованого програмування [230]. Проте доцільно проаналізувати можливість застосування інших парадигм програмування.

### **1.3. Аналіз парадигм програмування для розроблення програмного забезпечення систем обробки темпоральних мультимодальних даних**

Розглянемо парадигми програмування, які певною мірою стосуються обробки темпоральних мультимодальних даних, та оцінімо можливість їх застосування як основи процесу розроблення програмного забезпечення для ефективної обробки темпоральних мультимодальних даних.

*Реактивне програмування* (reactive programming) [231] – це парадигма програмування, яка ґрунтується на понятті *асинхронного потоку даних*. Зміни стану системи, що викликаються вхідними асинхронними потоками даних, автоматично розповсюджуються через мережу залежних обчислень за базовою моделлю оброблення даних. Під *поток*ом (event) мається на увазі послідовність упорядкованих подій на часовій шкалі: дані, що вводяться користувачем з клавіатури, події маніпулятора «миша», запити даних тощо. Події характеризуються дискретністю – вони відбуваються в окремі моменти часу (наприклад, натискання клавіш на клавіатурі, зміна місця розташування об'єкта тощо). Другим базовим поняттям парадигми реактивного програмування є *поведінка* (behaviour). Поведінка позначає



змінні за часом значення. Прикладом поведінки є час. Більшість мов реактивного програмування передбачають специфічні примітиви для подання значень часу. Парадигма реактивного програмування може бути реалізована за допомогою багатьох мов програмування, включаючи мови Haskell [232] та Scala [233].

*Програмування потоків даних* (dataflow programming) [234, 235] являє собою парадигму програмування, яка внутрішньо представляє застосунок як орієнтований граф. Застосунок є множиною *вузлів* з вхідними та вихідними портами. Ці вузли можуть бути або *джерелами даних* (sources), або *приймачами даних* (sinks), або *блоками обробки даних* (processing blocks). Вузли зв'язані спрямованими ребрами, які позначають потоки даних між ними. Застосунок являє собою композицію блоків обробки даних з одним або декількома джерелами даних та одним або декількома приймачами даних.

Блок обробки даних активується, коли на його вхідні порти надходять дані з попереднього вузла. По завершенню обробки дані передаються з вихідного порту блоку обробки на наступний вузол. Отже, вузол стає активованим, коли на його входи з'являються дані. При цьому дані у графі передаються в одному напрямку. Така схема обчислень робить можливим паралельне обчислення у конвеєрний спосіб.

Більшість мов візуального програмування, які використовують блочну архітектуру для представлення *потoku обробки даних* (work flow), ґрунтуються на використанні парадигми програмування потоків даних. Прикладами можуть слугувати мова візуального програмування середовища LabVIEW [236] та мова програмування MVPL [237].

Іншими прикладами мов програмування, які дозволяють реалізувати парадигму програмування потоків даних, є VHDL [238] та Signal [239].

Окремим випадком парадигми програмування потоків даних є *програмування на основі потоку* (Flow-Based Programming) [240].

Парадигма програмування на основі потоку визначає застосунок як мережу процесів, які передають один одному фрагменти даних (так звані *інформаційні пакети*), що проходять через заздалегідь визначені з'єднання. Передбачається, що послідовність з'єднання процесів у мережу може довільно змінюватись без зміни сутності самих процесів з метою розроблення нових застосунків для інших випадків обробки даних. Таким чином, програмування на основі потоку реалізує компоненто-орієнтований підхід до програмування.

Основними характерними особливостями програмування на основі потоку є використання асинхронних, паралельних процесів та робота з інформаційними пакетами, що мають визначений час існування.

Парадигма програмування на основі потоку може бути реалізована за допомогою мов програмування Java (JavaFBP, JavaFBP Web Sockets), C# (C#FBP), C++ (CppFBP), JavaScript (JSFBP) тощо.

*Подіє-орієнтоване програмування* (event-driven programming) [241] є парадигмою програмування, згідно з якою потік виконання програмного застосунку визначається подіями. Відповідно застосунок, що є керованим подіями, має виявляти події під час їх виникнення та реагувати на них за допомогою відповідної процедури обробки подій. До подій відносяться дії користувача, повідомлення операційної системи або іншої програми.

Подіє-орієнтоване програмування реалізує стратегію проектування керуючого потоку програми, яка має реагувати на події, зовнішні для самої програми. Такі програми можуть використовувати потоки, підпрограми, генератори та інші засоби для досягнення мети обробки зовнішніх подій.

Центральним елементом програми, що є керованою подіями, є планувальник, який отримує потік подій і передає кожну подію відповідному обробнику подій. Планувальник залишається активним, доки не отримує подію, яка спричинить припинення роботи програми.

Ця парадигма програмування використовується для програмування графічних інтерфейсів користувача і розроблення веб-застосунків та може бути реалізована за допомогою будь-якої високорівневої мови програмування. Прикладами мов програмування, які підтримують парадигму подіє-орієнтованого програмування, є C# та JavaScript.

*Програмування, кероване даними* (data-driven programming) [242] є парадигмою програмування, в якій при розробленні програмного застосунку, відбувається розмежування програмного коду та структури даних, для обробки яких він призначений. Мета цього підходу – надати розробнику можливість зміни логіки програми шляхом зміни її коду, а не структури даних. Характерною особливістю програмування, керованого даними, є те, що дані в цій парадигмі є не просто станом певного об'єкта, а вони фактично визначають потік управління програмою. Прикладами мов програмування, які підтримують парадигму програмування, керованого даними, є AWK [243, 244] та Clojure [245].

*Процесно-орієнтоване програмування* (process-oriented programming) [246–248] можна визначити як парадигму, яка ґрунтується на визначенні структури процесу та зв'язку між процесами системи. В основі цієї парадигми лежить теорія взаємодії послідовних процесів (communicating sequential processes) Хоара [249].

Парадигма процесно-орієнтованого програмування передбачає ідентифікацію процесів, необхідних для вирішення задачі, в тому числі з точки зору використання ресурсів обчислювальної системи. Прикладами мов програмування, які підтримують парадигму процесно-орієнтованого програмування є ProcessJ [250], ossam-π [251], Rain [252].

*Програмування систем реального часу* (real-time programming) [253], як парадигма програмування, ґрунтується на понятті *системи реального часу* – системи, правильність поведінки якої залежить від логічних результатів обчислень та фізичного часу, коли ці результати отримуються,

з метою забезпечення відповідності швидкості обчислень динаміці фізичного процесу, для контролю якого або управління яким ці обчислення виконуються.

Особливість програмування систем реального часу полягає у тому, що логіка виконання програми визначається зовнішніми подіями (наприклад, сигналами, що надходять від давачів) та може явно залежати від часу. Зокрема, можуть висуватись жорсткі часові обмеження, які означають, що неможливість обчислити результат за певний час вважається помилкою у роботі системи. Отже, однією з найбільш важливих особливостей систем реального часу є те, що час реакції на вхідні сигнали має задовольняти заданим обмеженням.

Прикладом спеціалізованих мов програмування, які застосовуються для програмування систем реального часу, можуть слугувати мови Esterel [254, 255] та Lustre [256, 257].

*Інтерактивне програмування* (interactive programming, on-the-fly programming) [258, 259] є відносно новою парадигмою програмування, згідно з якою програміст може вносити зміни у програмний код під час його виконання. Ця парадигма може застосовуватись у випадках, коли об'єкт моніторингу або управління швидко змінюється. Прикладами застосування цієї парадигми можуть слугувати обробка результатів експериментальних досліджень у реальному часі [258] та обробка аудіо сигналу у процесі виконання музичного твору [259].

Парадигма інтерактивного програмування може бути реалізована за допомогою мультипарадигмових мов, наприклад, мови програмування JavaScript, та спеціалізованої мови програмування ChucK [260].

Аналіз розглянутих парадигм програмування дозволяє стверджувати, що жодна з них не надає всіх можливостей, необхідних для ефективної обробки темпоральних мультимодальних даних цифрових двійників, оскільки не підтримує одночасно всі характерні особливості обробки даних

такого виду: оперування поняттям часу, обробка потоків даних, визначення структур мультимодальних даних (табл. 1.2).

Частково ці вимоги задовольняють реактивне програмування та програмування систем реального часу, оскільки ці парадигми підтримує обробку темпоральних значень та потоків даних. Проте доцільним є розроблення специфічної парадигми програмування, яка буде враховувати всі особливості процесу створення цифрового двійника та обробки його даних.

Таблиця 1.2 – Порівняння парадигм програмування

Парадигма програмування	Оперування поняттям часу	Обробка потоків даних	Визначення структур мультимодальних даних
Реактивне програмування	так	так	ні
Програмування потоків даних	ні	так	ні
Подіє-орієнтоване програмування	опосередковано	так	ні
Програмування, кероване даними	ні	так	ні
Процесно-орієнтоване програмування	опосередковано	так	ні
Програмування систем реального часу	так	так	ні
Інтерактивне програмування	опосередковано	ні	ні

Створення нової парадигми програмування, яка призначена для розроблення програмного забезпечення систем обробки темпоральних мультимодальних даних цифрових двійників, дозволить спростити процес розроблення нового програмного забезпечення на основі технології цифрових двійників та підвищити ефективність обробки темпоральних мультимодальних даних.

#### **1.4. Висновки до розділу 1**

Проведений аналіз сучасного стану вирішення проблеми подання та обробки мультимодальних даних цифрових двійників досліджуваних об'єктів дозволяє зробити загальний висновок про необхідність і доцільність розроблення нових методів, моделей та засобів для вирішення задачі подання та обробки темпоральних мультимодальних даних цифрового двійника для опису досліджуваного об'єкта, які дозволять:

- підвищити ефективність обробки темпоральних мультимодальних даних цифрових двійників;
- спростити розроблення програмного забезпечення систем обробки даних цифрових двійників;
- розширити коло застосувань технології цифрових двійників.

До основних недоліків існуючих методів, моделей та програмних засобів обробки даних цифрових двійників можна віднести наступне.

1. На сучасному етапі розвитку технології цифрових двійників відсутній загальний, універсальний підхід до вирішення задачі створення цифрового двійника довільного досліджуваного об'єкта.
2. Існуючі методи, моделі та програмні засоби обробки даних цифрових двійників сфокусовані на застосуванні технології

цифрових двійників у галузі виробництва, а також для вирішення задач оптимізації бізнес-процесів у великих організаційних структурах (підприємствах, транспортних компаніях, стадіонах, містах, енергетичних системах тощо).

3. Візуальні моделі цифрових двійників та програмне забезпечення для їх відтворення забезпечують недостатній рівень реалістичності, оскільки ґрунтуються лише на графічному поданні фізичних об'єктів та не враховують інформацію інших модельностей, що сприймається органами чуття людини.

В результаті проведеного аналізу визначено напрями подальшого дослідження з метою усунення виявлених недоліків існуючих методів, моделей і програмних засобів обробки даних цифрових двійників, підвищення ефективності обробки темпоральних мультимодальних даних у комп'ютерних системах та спрощення процесу розроблення програмного забезпечення систем обробки даних цифрових двійників. Зокрема, проведене дослідження дозволило зробити такі висновки:

1. Технологія цифрових двійників може бути застосована для опису довільного досліджуваного об'єкта. При цьому доцільно створювати цифрові двійники, які можуть бути класифіковані:
  - за призначенням, як цифровий двійник-екземпляр та цифровий двійник-агрегат;
  - за рівнем інтеграції, як цифрова тінь;
  - за рівнем складності, як цифровий двійник другого, третього та четвертого рівнів.
2. Технологія мультимедіа надає необхідні можливості, які можуть бути застосовані для створення реалістичних візуальних моделей цифрових двійників.

3. Доцільним є розроблення методів, моделей та програмних засобів обробки даних цифрових двійників довільних досліджуваних об'єктів.
4. Доцільним є розроблення парадигми програмування та моделей обчислення, спрямованих на ефективну обробку темпоральних мультимодальних даних цифрових двійників.
5. Доцільним є розроблення спеціалізованої мови програмування для розроблення програмного забезпечення систем обробки темпоральних мультимодальних, зокрема, мультимедійних, даних.
6. Доцільним є створення теоретичних основ для розроблення формальної специфікації цифрового двійника досліджуваного об'єкта.



## РОЗДІЛ 2. ТЕОРЕТИЧНІ ОСНОВИ ПОДАННЯ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВОГО ДВІЙНИКА ДЛЯ ОПИСУ ДОСЛІДЖУВАНОВОГО ОБ'ЄКТА

Для формальної специфікації цифрових двійників досліджуваних об'єктів потрібен теоретичний апарат, який забезпечить логіку подання та обробки темпоральних мультимодальних даних на рівні математичного, алгоритмічного та програмного забезпечення з урахуванням таких властивостей цих даних:

1. *Мультимодальність* – цифровий двійник визначається сукупністю даних різної природи; логіка подання та обробки даних цифрового двійника залежить від якісного та кількісного складу цієї сукупності даних.
2. *Темпоральність* – елементи сукупності даних цифрового двійника є впорядкованими за часом їх отримання (генерації, реєстрації); порядок слідування окремих елементів послідовності даних впливає на результат обробки всієї сукупності даних цифрового двійника.

Для опису цифрового двійника та ефективної обробки темпоральних мультимодальних даних потрібні специфічні математичні абстракції та механізми оперування ними. Сформулюємо основні вимоги до математичного апарату для опису та обробки даних цифрового двійника:

1. Можливість подання сукупності даних як структури взаємопов'язаних елементів.
2. Можливість врахування порядку слідування елементів сукупності даних при виконанні логічних операції над ними.
3. Можливість перевпорядковування елементів сукупності даних.

Цим вимогам задовольняє *алгебраїчна система агрегатів* [1, 3, 12, 13, 21]. Розглянемо її основні положення та можливості щодо подання та обробки темпоральних мультимодальних даних цифрових двійників.

## 2.1. Алгебраїчна система агрегатів як інструмент формального опису досліджуваного об'єкта

Для визначення основних положень алгебраїчної системи агрегатів сформулюємо деякі базові поняття.

*Алгебраїчна система* [261] – це сукупність множин  $(\mathcal{M}, \mathcal{F}, \mathcal{R})$ , що задовольняють деякій системі аксіом, де  $\mathcal{M}$  – це непуста множина (носій), елементи якої є елементами системи;  $\mathcal{F}$  – множина операцій;  $\mathcal{R}$  – множина відношень. Алгебраїчна система з пустою множиною відношень називається *алгеброю*, а система з пустою множиною операцій – *моделлю*.

*Кортежем* [261–264] називають впорядковану скінчену сукупність елементів, що можуть повторюватись.

Тоді, *алгебраїчна система агрегатів* (АСА) – це алгебраїчна система, носієм якої є непуста множина об'єктів, які називатимемо *агрегатами*.

Визначення 2.1. *Агрегат*  $A$  – це впорядкована скінченна сукупність елементів, яка визначається кортежем множин  $\{A\}$  та кортежем кортежів елементів  $\langle A \rangle$ , причому елементи  $a_i^j$  кожного кортежу елементів  $\langle a_i^j \rangle_{i=1}^{n_j} \in \langle A \rangle$  належать до відповідної множини  $M_j \in \{A\}$ ,  $j = [1 \dots N]$ , що задає взаємо-однозначний зв'язок між порядком слідування множин та порядком слідування кортежів елементів:

$$A = \llbracket M_j | \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N = \llbracket \{A\} | \langle A \rangle \rrbracket, \quad (2.1)$$

де  $\{A\}$  – кортеж множин  $M_j$ ;  $\langle A \rangle$  – кортеж кортежів елементів  $\langle a_i^j \rangle_{i=1}^{n_j}$ ;  $a_i^j$  – окремий елемент (значення) або складений елемент (кортеж

однорідних значень або кортеж різнорідних значень),  $a_i^j \in M_j$ .

Визначальними рисами агрегату, які відрізняють цю математичну абстракцію від інших, є наступне:

- агрегат є складеним математичним об'єктом, всі компоненти якого є впорядкованими;
- елементами кортежів, з яких складається агрегат, можуть бути окремі значення або кортежі значень, причому кортежі значень можуть складатись як зі значень одного типу, так і зі значень різних типів; при цьому кожен кортеж містить елементи, що належать одній множині.

Згідно з формулою (2.1) елементи першого кортежу належать до першої множини, елементи другого кортежу належать до другої множини тощо. Порядок слідування множин у агрегаті визначає, як будуть виконуватися операції над агрегатом. Множини у кортежі множин можуть повторюватись; це означає, що в агрегат входять декілька кортежів, що складаються з елементів одного типу.

Окремий елемент кортежу може бути чітким значенням  $a_i$ , нечітким значенням  $\tilde{a}_i$  або може бути невизначеним  $\_$ . Відмінність невизначеного елемента від визначених (чіткого та нечіткого значень) полягає у тому, що невизначений елемент не має значення або воно невідомо до певного моменту, який визначається семантично.

Кортеж, який містить лише невизначені елементи називатимемо *невизначеним* та позначатимемо  $\langle \_ \rangle$ . Кортеж, який не містить жодного є *пустим*; позначатимемо його  $\langle \emptyset \rangle$ .

Визначення 2.2. Агрегат, який складається лише з пустих кортежів, називатимемо *пустим*:

$$A = \llbracket M_j | \langle \emptyset \rangle \rrbracket_{j=1}^N \quad (2.2)$$

Визначення 2.3. Агрегат  $A_\emptyset$ , який не містить жодного компоненту, називатимемо *нуль-агрегатом*:

$$A_\emptyset = \llbracket \emptyset | \langle \emptyset \rangle \rrbracket \quad (2.3)$$

Нуль-агрегат відіграє у АСА роль *нейтрального елементу*.

Визначення 2.4. Агрегат, який складається лише з невизначених кортежів називатимемо *невизначеним*:

$$A = \llbracket M_j | \langle \_ \rangle \rrbracket_{j=1}^N. \quad (2.4)$$

Невизначений агрегат відрізняється від пустого агрегата тим, що його елементи отримують значення при виконанні певної умови. В практичному сенсі це може означати, що дані будуть отримані з наперед визначеного давача (тобто тип даних є відомим), коли цей давач буде увімкнено або з ним буде встановлено зв'язок.

*Довжина* кортежу  $|\bar{a}|$  – це кількість елементів у цьому кортежі.

*Довжина* агрегата  $|A|$  – це кількість кортежів в цьому агрегаті. Довжина нуль-агрегата (2.3) дорівнює нулю:  $|A_\emptyset| = 0$ . Довжина пустого агрегата (2.2) складає  $|A| = N$ . Довжина невизначеного агрегата (2.4) також складає  $|A| = N$ .

*Сукупна довжина* агрегата  $||A||$  – це сума довжин кортежів агрегата. Сукупна довжина нуль-агрегата дорівнює нулю:  $||A_\emptyset|| = 0$ .

Порядок слідування множин і відповідних їм кортежів в агрегаті є важливим – саме він визначає, як будуть виконуватись операції над агрегатами. Визначимо це через поняття *сумісності* агрегатів.

Визначення 2.5. Агрегати  $A_1$  та  $A_2$  називаються *сумісними* ( $A_1 \doteq A_2$ ), якщо вони мають однакову довжину, а тип та порядок слідування множин в них збігаються, тобто виконуються умови:

$$\begin{cases} |A_1| = |A_2| \\ \{A_1\} \equiv \{A_2\}. \end{cases} \quad (2.5)$$

Наприклад, агрегати, визначені формулами (2.6), є сумісними.

$$\begin{aligned} A_1 &= \llbracket M_1, M_2, M_3 | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1^1}, \langle a_{i_2}^1 \rangle_{i_2=1}^{n_2^1}, \langle a_{i_3}^1 \rangle_{i_3=1}^{n_3^1} \rrbracket, \\ A_2 &= \llbracket M_1, M_2, M_3 | \langle a_{i_1}^2 \rangle_{i_1=1}^{n_1^2}, \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2^2}, \langle a_{i_3}^2 \rangle_{i_3=1}^{n_3^2} \rrbracket. \end{aligned} \quad (2.6)$$

Визначення 2.6. Агрегати  $A_1$  та  $A_2$  називаються *квазісумісними* ( $A_1 \doteq A_2$ ) якщо тип та порядок слідування множин в них збігаються частково, при цьому немає вимоги щодо рівності довжин цих агрегатів, тобто виконуються умови:

$$\begin{cases} \{A_1\} \not\equiv \{A_2\} \\ \{A_1\} \cap \{A_2\} \neq \emptyset. \end{cases} \quad (2.7)$$

Наприклад, агрегати, визначені формулами (2.8), є квазісумісними.

$$\begin{aligned} A_1 &= \llbracket M_1, M_2, M_3^1 | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1^1}, \langle a_{i_2}^1 \rangle_{i_2=1}^{n_2^1}, \langle a_{i_3}^1 \rangle_{i_3=1}^{n_3^1} \rrbracket, \\ A_2 &= \llbracket M_1, M_2, M_3^2 | \langle a_{i_1}^2 \rangle_{i_1=1}^{n_1^2}, \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2^2}, \langle a_{i_3}^2 \rangle_{i_3=1}^{n_3^2} \rrbracket. \end{aligned} \quad (2.8)$$

Визначення 2.7. Агрегати  $A_1$  та  $A_2$  називаються *несумісними* ( $A_1 \doteq A_2$ ), якщо тип та порядок слідування множин в них не збігаються, тобто виконується умова:

$$\{A_1\} \cap \{A_2\} = \emptyset \quad (2.9)$$

Наприклад, агрегати, визначені формулами (2.10), є несумісними.

$$\begin{aligned} A_1 &= \llbracket M_1^1, M_2^1, M_3^1 | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1^1}, \langle a_{i_2}^1 \rangle_{i_2=1}^{n_2^1}, \langle a_{i_3}^1 \rangle_{i_3=1}^{n_3^1} \rrbracket, \\ A_2 &= \llbracket M_1^2, M_2^2, M_3^2 | \langle a_{i_1}^2 \rangle_{i_1=1}^{n_1^2}, \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2^2}, \langle a_{i_3}^2 \rangle_{i_3=1}^{n_3^2} \rrbracket. \end{aligned} \quad (2.10)$$

Особливим випадком несумісності є *прихована сумісність*.

Визначення 2.8. Агрегати  $A_1$  та  $A_2$  називаються *приховано сумісними*,  $A_1 (\div) A_2$ , якщо обидва агрегати мають однаковий набір множин, але їх порядок слідування відрізняється, тобто виконуються умови:

$$\begin{cases} \{A_1\} \not\equiv \{A_2\} \\ |A_1| = |A_2| = N \\ \forall M_j \subset \{A_k\}, \end{cases} \quad (2.11)$$

де  $j = [1, \dots, N]$ ,  $k = [1, 2]$ .

Наприклад, агрегати, визначені формулами (2.12), є приховано сумісними.

$$\begin{aligned} A_1 &= \llbracket M_1, M_2, M_3 | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1^1}, \langle a_{i_2}^1 \rangle_{i_2=1}^{n_2^1}, \langle a_{i_3}^1 \rangle_{i_3=1}^{n_3^1} \rrbracket, \\ A_2 &= \llbracket M_2, M_3, M_1 | \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2^2}, \langle a_{i_3}^2 \rangle_{i_3=1}^{n_3^2}, \langle a_{i_1}^2 \rangle_{i_1=1}^{n_1^2} \rrbracket. \end{aligned} \quad (2.12)$$

Приховано сумісні агрегати можуть бути перетворені на сумісні шляхом застосування до них певних операцій.

## 2.2. Операції та відношення в алгебраїчній системі агрегатів

Операції над агрегатами включають операції таких категорій:

1. Логічні операції
2. Операції впорядкування
3. Арифметичні операції.

Розглянемо ці операції докладно.

### 2.2.1. Логічні операції над агрегатами

Логічними операціями [1, 13, 21] над агрегатами є наступні операції: *об'єднання* ( $\cup$ ), *переріз* ( $\cap$ ), *різниця* ( $\setminus$ ), *симетрична різниця* ( $\Delta$ ) та *виключний переріз* ( $\neg$ ). Визначимо ці операції.

*Об'єднанням* двох агрегатів  $A_1$  та  $A_2$  є агрегат  $B$ , який містить елементи кортежів, які належать обом агрегатам та впорядковані за таким правилом:

1. Якщо  $A_1 \div A_2$  та

$$A_1 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1^1, b_2^1, \dots, b_m^1 \rangle, \dots, \langle w_1^1, w_2^1, \dots, w_n^1 \rangle \rrbracket,$$
$$A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle b_1^2, b_2^2, \dots, b_q^2 \rangle, \dots, \langle w_1^2, w_2^2, \dots, w_p^2 \rangle \rrbracket,$$

то елементи  $i$ -того кортежу агрегата  $A_2$  додаються у кінець  $i$ -го кортежу агрегата  $A_1$ :

$$B = A_1 \cup A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^1, a_2^1, \dots, a_l^1, a_1^2, a_2^2, \dots, a_r^2 \rangle, \quad (2.13)$$
$$\langle b_1^1, b_2^1, \dots, b_m^1, b_1^2, b_2^2, \dots, b_q^2 \rangle, \dots, \langle w_1^1, w_2^1, \dots, w_n^1, w_1^2, w_2^2, \dots, w_p^2 \rangle \rrbracket.$$

2. Якщо  $A_1 \subseteq A_2$  та

$$A_1 = \llbracket M_1^1, M_2^1, \dots, M_N^1 | \langle a_1, a_2, \dots, a_l \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle \rrbracket,$$

$$A_2 = \llbracket M_1^2, M_2^2, \dots, M_K^2 | \langle c_1, c_2, \dots, c_r \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket,$$

то кортеж кортежів агрегата  $A_2$  додається у кінець кортежу кортежів агрегата  $A_1$  та кортеж множин агрегата  $A_2$  додається у кінець кортежу множин агрегата  $A_1$ :

$$\begin{aligned} B = A_1 \cup A_2 = \llbracket M_1^1, M_2^1, \dots, M_N^1, M_1^2, M_2^2, \dots, M_K^2 | \langle a_1, a_2, \dots, a_l \rangle, \\ \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle, \langle c_1, c_2, \dots, c_r \rangle, \\ \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket. \end{aligned} \quad (2.14)$$

3. Якщо  $A_1 \doteq A_2$  та

$$\begin{aligned} A_1 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1 | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_1^1, f_2^1, \dots, f_t^1 \rangle, \dots, \\ \langle w_1, w_2, \dots, w_n \rangle \rrbracket, \\ A_2 = \llbracket M_1, M_2^2, \dots, M_x, \dots, M_K^2 | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle f_1^2, f_2^2, \dots, f_v^2 \rangle, \dots, \\ \langle z_1, z_2, \dots, z_p \rangle \rrbracket, \end{aligned}$$

то:

- а) елементи  $i$ -того кортежу агрегата  $A_2$  додаються у кінець  $i$ -го кортежу агрегата  $A_1$ , якщо елементи цих кортежів належать до тієї ж самої  $i$ -тої множини;
- б) для всіх  $i$ -тих кортежів, елементи яких належать до різних множин, кортеж кортежів агрегата  $A_2$  додається у кінець кортежу кортежів агрегата  $A_1$  та кортеж множин агрегата  $A_2$  додається у кінець кортежу множин агрегата  $A_1$ , за винятком кортежів, які підпадають під дію правила а) та виключаються з кортежу кортежів:

$$\begin{aligned} B = A_1 \cup A_2 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1, M_2^2, \dots, M_K^2 | \langle a_1^1, a_2^1, \dots, a_l^1, a_1^2, \\ a_2^2, \dots, a_r^2 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_1^1, f_2^1, \dots, f_t^1, f_1^2, f_2^2, \dots, f_v^2 \rangle, \dots, \\ \langle w_1, w_2, \dots, w_n \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket. \end{aligned} \quad (2.15)$$



Перерізом двох агрегатів  $A_1$  та  $A_2$  є агрегат  $B$ , який містить компоненти, які є спільними для цих агрегатів, впорядковані за таким правилом:

1. Якщо  $A_1 \div A_2$  та

$$A_1 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1^1, b_2^1, \dots, b_m^1 \rangle, \dots, \langle w_1^1, w_2^1, \dots, w_n^1 \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle b_1^2, b_2^2, \dots, b_q^2 \rangle, \dots, \langle w_1^2, w_2^2, \dots, w_p^2 \rangle \rrbracket,$$

то  $i$ -тий кортеж агрегата  $B$  містить спільні елементи  $i$ -тих кортежів обох агрегатів зі збереженням порядку слідування цих елементів:

$$B = A_1 \cap A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_{l_1}^1, \dots, a_{l_\alpha}^1, a_{r_1}^2, \dots, a_{r_\beta}^2 \rangle, \langle b_{m_1}^1, \dots, b_{m_\gamma}^1, b_{q_1}^2, \dots, b_{q_\delta}^2 \rangle, \dots, \langle w_{n_1}^1, \dots, w_{n_\lambda}^1, w_{p_1}^2, \dots, w_{p_\mu}^2 \rangle \rrbracket, \quad (2.16)$$

де  $a_{l_i}^1 \in \overline{a^1}, a_{l_i}^1 \in \overline{a^2}, i \in \langle 1, \dots, \alpha \rangle; a_{r_j}^2 \in \overline{a^1}, a_{r_j}^2 \in \overline{a^2}, j \in \langle 1, \dots, \beta \rangle;$

$b_{m_k}^1 \in \overline{b^1}, b_{m_k}^1 \in \overline{b^2}, k \in \langle 1, \dots, \gamma \rangle; b_{q_s}^2 \in \overline{b^1}, b_{q_s}^2 \in \overline{b^2}, s \in \langle 1, \dots, \delta \rangle;$

$w_{n_u}^1 \in \overline{w^1}, w_{n_u}^1 \in \overline{w^2}, u \in \langle 1, \dots, \lambda \rangle; w_{p_y}^2 \in \overline{w^1}, w_{p_y}^2 \in \overline{w^2}, y \in \langle 1, \dots, \mu \rangle.$

Це означає наступне. Будемо проходити по елементах кортежу  $\overline{a^1} = \langle a_1^1, a_2^1, \dots, a_l^1 \rangle$  від його першого елементу  $a_1^1$  до останнього елементу  $a_l^1$  та порівнювати кожен наступний елемент  $a_{l_i}^1$  ( $1 \leq l_i \leq l$ , де  $l = |\overline{a^1}|$ ) з елементами кортежу  $\overline{a^2} = \langle a_1^2, a_2^2, \dots, a_r^2 \rangle$ . Тоді  $a_{l_1}^1$  – це перший елемент кортежу  $\overline{a^1}$ , який також присутній у кортежу  $\overline{a^2}$ , та  $a_{l_\alpha}^1$  – це останній елемент кортежу  $\overline{a^1}$ , який також присутній у кортежу  $\overline{a^2}$ . Аналогічно, будемо проходити по елементах кортежу  $\overline{a^2}$  від його першого елементу  $a_1^2$  до останнього елементу  $a_r^2$  та порівнювати кожен наступний елемент  $a_{r_j}^2$  ( $1 \leq r_j \leq r$ , де  $r = |\overline{a^2}|$ ) з елементами кортежу  $\overline{a^1}$ . Тоді  $a_{r_1}^2$  – це перший елемент кортежу  $\overline{a^2}$ , який також присутній у кортежу  $\overline{a^1}$ , та  $a_{r_\beta}^2$  – це

останній елемент кортежу  $\overline{a^2}$ , який також присутній у кортежу  $\overline{a^1}$ . Отже, результатом перерізу кортежів  $\overline{a^1}$  та  $\overline{a^2}$  є кортеж  $\langle a_{l_1}^1 \dots a_{l_\alpha}^1, a_{r_1}^2 \dots a_{r_\beta}^2 \rangle$ .

Наприклад, якщо  $\overline{a^1} = \langle 9, 5, \mathbf{2}, 7, 9, \mathbf{3}, 6 \rangle$  та  $\overline{a^2} = \langle 1, \mathbf{3}, \mathbf{2}, 4, \mathbf{7}, \mathbf{3}, 8 \rangle$ , то  $a_{l_1}^1 = \mathbf{2}$  та  $a_{l_\alpha}^1 = \mathbf{3}$ . Водночас,  $a_{r_1}^2 = \mathbf{3}$  та  $a_{r_\beta}^2 = \mathbf{3}$ . Отже,  $\overline{a^1} \cap \overline{a^2} = \langle \mathbf{2}, \mathbf{7}, \mathbf{3}, \mathbf{3}, \mathbf{2}, \mathbf{7}, \mathbf{3} \rangle$ .

2. Якщо  $A_1 \doteq A_2$ , то  $B = A_1 \cap A_2 = \llbracket \emptyset | \langle \emptyset \rangle \rrbracket = A_\emptyset$ .

3. Якщо  $A_1 \doteq A_2$  та

$$A_1 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1 | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_1^1, f_2^1, \dots, f_t^1 \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2^2, \dots, M_x, \dots, M_K^2 | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle f_1^2, f_2^2, \dots, f_v^2 \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket,$$

то агрегат  $B$  містить спільні елементи обох агрегатів лише для кортежів, які є сумісними; несумісні кортежі відкидаються, отже, кількість кортежів скорочується:

$$B = A_1 \cap A_2 = \llbracket M_1, \dots, M_x | \langle a_{l_1}^1, \dots, a_{l_\alpha}^1, a_{r_1}^2, \dots, a_{r_\beta}^2 \rangle, \dots, \langle f_{t_1}^1, \dots, f_{t_\rho}^1, f_{v_1}^2, \dots, f_{v_\omega}^2 \rangle \rrbracket \quad (2.17)$$

Різницею агрегатів  $A_1$  та  $A_2$  є агрегат  $B$ , який містить компоненти першого агрегата, яких немає у другому агрегаті, впорядковані за таким правилом:

1. Якщо  $A_1 \div A_2$  та

$$A_1 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1^1, b_2^1, \dots, b_m^1 \rangle, \dots, \langle w_1^1, w_2^1, \dots, w_n^1 \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle b_1^2, b_2^2, \dots, b_q^2 \rangle, \dots, \langle w_1^2, w_2^2, \dots, w_p^2 \rangle \rrbracket,$$

то  $i$ -тий кортеж агрегата  $B$  містить елементи  $i$ -того кортежу агрегата  $A_1$ , які відсутні у  $i$ -тому кортежі агрегата  $A_2$ :

$$B = A_1 \setminus A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_{l_1}^1, \dots, a_{l_\alpha}^1 \rangle, \langle b_{m_1}^1, \dots, b_{m_\gamma}^1 \rangle, \dots, \langle w_{n_1}^1, \dots, w_{n_\lambda}^1 \rangle \rrbracket, \quad (2.18)$$

де  $a_{l_i}^1 \in \overline{a^1}, a_{l_i}^1 \notin \overline{a^2}, i \in \langle 1, \dots, \alpha \rangle; b_{m_k}^1 \in \overline{b^1}, b_{m_k}^1 \notin \overline{b^2}, k \in \langle 1, \dots, \gamma \rangle; w_{n_u}^1 \in \overline{w^1}, w_{n_u}^1 \notin \overline{w^2}, u \in \langle 1, \dots, \lambda \rangle$ .

2. Якщо  $A_1 \doteq A_2$ , то  $B = A_1 \setminus A_2 = A_1$ .

3. Якщо  $A_1 \doteq A_2$  та

$$A_1 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1 | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_1^1, f_2^1, \dots, f_t^1 \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2^2, \dots, M_x, \dots, M_K^2 | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle f_1^2, f_2^2, \dots, f_v^2 \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket,$$

то  $i$ -тий кортеж агрегата  $B$  містить лише елементи  $i$ -того кортежу агрегата  $A_1$ , які відсутні у  $i$ -тому кортежі агрегата  $A_2$ , якщо цей  $i$ -тий кортеж є сумісним для обох агрегатів, та містить всі елементи  $i$ -того кортежу агрегата  $A_1$ , якщо  $i$ -тий кортеж є несумісним:

$$B = A_1 \setminus A_2 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1 | \langle a_{i_1}^1, \dots, a_{i_\alpha}^1 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_{t_1}^1, \dots, f_{t_\rho}^1 \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle \rrbracket, \quad (2.19)$$

де  $a_{l_i}^1 \in \overline{a^1}, a_{l_i}^1 \notin \overline{a^2}, i \in \langle 1, \dots, \alpha \rangle; f_{t_e}^1 \in \overline{f^1}, f_{t_e}^1 \notin \overline{f^2}, e \in \langle 1, \dots, \rho \rangle$ .

Симетричною різницею агрегатів  $A_1$  та  $A_2$  є агрегат  $B$ , який містить компоненти першого агрегата, яких немає у другому агрегаті, та компоненти другого агрегата, яких немає у першому агрегаті, впорядковані за таким правилом:

1. Якщо  $A_1 \div A_2$  та

$$A_1 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1^1, b_2^1, \dots, b_m^1 \rangle, \dots, \langle w_1^1, w_2^1, \dots, w_n^1 \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle b_1^2, b_2^2, \dots, b_q^2 \rangle, \dots, \langle w_1^2, w_2^2, \dots, w_p^2 \rangle \rrbracket,$$

то  $i$ -тий кортеж агрегата  $B$  містить елементи  $i$ -того кортежу агрегата  $A_1$ , які відсутні у  $i$ -тому кортежі агрегата  $A_2$ , та елементи  $i$ -того кортежу агрегата  $A_2$ , які відсутні у  $i$ -тому кортежі агрегата  $A_1$ :

$$B = A_1 \Delta A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_{l_1}^1, \dots, a_{l_\alpha}^1, a_{r_1}^2, \dots, a_{r_\beta}^2 \rangle, \langle b_{m_1}^1, \dots, b_{m_\gamma}^1, b_{q_1}^2, \dots, b_{q_\delta}^2 \rangle, \dots, \langle w_{n_1}^1, \dots, w_{n_\lambda}^1, w_{p_1}^2, \dots, w_{p_\mu}^2 \rangle \rrbracket, \quad (2.20)$$

де  $a_{l_i}^1 \in \overline{a^1}, a_{l_i}^1 \notin \overline{a^2}, i \in \langle 1, \dots, \alpha \rangle; a_{r_j}^2 \notin \overline{a^1}, a_{r_j}^2 \in \overline{a^2}, j \in \langle 1, \dots, \beta \rangle;$

$b_{m_k}^1 \in \overline{b^1}, b_{m_k}^1 \notin \overline{b^2}, k \in \langle 1, \dots, \gamma \rangle; b_{q_s}^2 \notin \overline{b^1}, b_{q_s}^2 \in \overline{b^2}, s \in \langle 1, \dots, \delta \rangle;$

$w_{n_u}^1 \in \overline{w^1}, w_{n_u}^1 \notin \overline{w^2}, u \in \langle 1, \dots, \lambda \rangle; w_{p_y}^2 \notin \overline{w^1}, w_{p_y}^2 \in \overline{w^2}, y \in \langle 1, \dots, \mu \rangle.$

2. Якщо  $A_1 \ominus A_2$  та

$$A_1 = \llbracket M_1^1, M_2^1, \dots, M_N^1 | \langle a_1, a_2, \dots, a_l \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle \rrbracket,$$

$$A_2 = \llbracket M_1^2, M_2^2, \dots, M_K^2 | \langle c_1, c_2, \dots, c_r \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket,$$

то агрегат  $B$  дорівнює результату об'єднання агрегатів  $A_1$  та  $A_2$ :

$$B = A_1 \Delta A_2 = \llbracket M_1^1, M_2^1, \dots, M_N^1, M_1^2, M_2^2, \dots, M_K^2 | \langle a_1, a_2, \dots, a_l \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle, \langle c_1, c_2, \dots, c_r \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket = A \quad (2.21)$$

3. Якщо  $A_1 \div A_2$  та

$$A_1 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1 | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_1^1, f_2^1, \dots, f_t^1 \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2^2, \dots, M_x, \dots, M_K^2 | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle f_1^2, f_2^2, \dots, f_v^2 \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket,$$

то  $i$ -тий кортеж агрегата  $B$  містить лише елементи  $i$ -того кортежу агрегата  $A_1$ , які відсутні у  $i$ -тому кортежі агрегата  $A_2$ , та елементи  $i$ -того кортежу агрегата  $A_2$ , які відсутні у  $i$ -тому кортежі агрегата  $A_1$ , якщо цей  $i$ -тий кортеж є сумісним для обох агрегатів, та містить всі елементи  $i$ -того кортежу агрегата  $A_1$  та всі елементи  $i$ -того кортежу агрегата  $A_2$ , якщо  $i$ -тий кортеж є несумісним:

$$B = A_1 \Delta A_2 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1, M_2^2, \dots, M_K^2 | \langle a_{l_1}^1, \dots, a_{l_\alpha}^1, a_{r_1}^2, \dots, a_{r_\beta}^2 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_{t_1}^1, \dots, f_{t_\rho}^1, f_{v_1}^2, \dots, f_{v_\omega}^2 \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket, \quad (2.22)$$

де  $a_{l_i}^1 \in \overline{a^1}, a_{l_i}^1 \notin \overline{a^2}, i \in \langle 1, \dots, \alpha \rangle; a_{r_j}^2 \notin \overline{a^1}, a_{r_j}^2 \in \overline{a^2}, j \in \langle 1, \dots, \beta \rangle;$   
 $f_{t_e}^1 \in \overline{f^1}, f_{t_e}^1 \notin \overline{f^2}, e \in \langle 1, \dots, \rho \rangle; f_{v_h}^2 \notin \overline{f^1}, f_{v_h}^2 \in \overline{f^2}, h \in \langle 1, \dots, \omega \rangle.$

*Виключним перерізом* двох агрегатів  $A_1$  та  $A_2$  є агрегат  $B$ , який містить лише компоненти агрегата  $A_1$ , які є спільними для цих агрегатів, впорядковані за таким правилом:

1. Якщо  $A_1 \div A_2$  та

$$A_1 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1^1, b_2^1, \dots, b_m^1 \rangle, \dots, \langle w_1^1, w_2^1, \dots, w_n^1 \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle b_1^2, b_2^2, \dots, b_q^2 \rangle, \dots, \langle w_1^2, w_2^2, \dots, w_p^2 \rangle \rrbracket,$$

то  $i$ -тий кортеж агрегата  $B$  містить елементи  $i$ -того кортежу агрегата  $A_1$ , які також містяться у  $i$ -тому кортежі агрегата  $A_2$ :

$$B = A_1 \neg A_2 = \llbracket M_1, M_2, \dots, M_N | \langle a_{l_1}^1, \dots, a_{l_\alpha}^1 \rangle, \langle b_{m_1}^1, \dots, b_{m_\gamma}^1 \rangle, \dots, \langle w_{n_1}^1, \dots, w_{n_\lambda}^1 \rangle \rrbracket, \quad (2.23)$$

де  $a_{l_i}^1 \in \overline{a^1}, a_{l_i}^1 \in \overline{a^2}, i \in \langle 1, \dots, \alpha \rangle; b_{m_k}^1 \in \overline{b^1}, b_{m_k}^1 \in \overline{b^2}, k \in \langle 1, \dots, \gamma \rangle;$   
 $w_{n_u}^1 \in \overline{w^1}, w_{n_u}^1 \in \overline{w^2}, u \in \langle 1, \dots, \lambda \rangle.$

2. Якщо  $A_1 \doteq A_2$ , то  $B = A_1 \neg A_2 = \llbracket \emptyset | \langle \emptyset \rangle \rrbracket = A_\emptyset.$

3. Якщо  $A_1 \doteq A_2$  та

$$A_1 = \llbracket M_1, M_2^1, \dots, M_x, \dots, M_N^1 | \langle a_1^1, a_2^1, \dots, a_l^1 \rangle, \langle b_1, b_2, \dots, b_m \rangle, \dots, \langle f_1^1, f_2^1, \dots, f_t^1 \rangle, \dots, \langle w_1, w_2, \dots, w_n \rangle \rrbracket,$$

$$A_2 = \llbracket M_1, M_2^2, \dots, M_x, \dots, M_K^2 | \langle a_1^2, a_2^2, \dots, a_r^2 \rangle, \langle d_1, d_2, \dots, d_q \rangle, \dots, \langle f_1^2, f_2^2, \dots, f_v^2 \rangle, \dots, \langle z_1, z_2, \dots, z_p \rangle \rrbracket,$$

то  $i$ -тий кортеж агрегата  $B$  містить елементи  $i$ -того кортежу агрегата  $A_1$ , які також містяться у  $i$ -тому кортежі агрегата  $A_2$ , якщо цей  $i$ -тий кортеж є сумісним для обох агрегатів, отже, кількість кортежів скорочується:

$$B = A_1 \neg A_2 = \llbracket M_1, \dots, M_x | \langle a_{l_1}^1, \dots, a_{l_\alpha}^1 \rangle, \dots, \langle f_{t_1}^1, \dots, f_{t_\rho}^1 \rangle \rrbracket, \quad (2.24)$$

де  $a_{l_i}^1 \in \overline{a^1}, a_{l_i}^1 \in \overline{a^2}, i \in \langle 1, \dots, \alpha \rangle; f_{t_e}^1 \in \overline{f^1}, f_{t_e}^1 \in \overline{f^2}, e \in \langle 1, \dots, \rho \rangle.$

Відмінність між операціями *виключного перерізу* та *перерізу* полягає у тому, що результатом *перерізу* агрегатів  $A_1$  та  $A_2$  є агрегат, який включає спільні компоненти обох агрегатів. Водночас, результатом *виключного перерізу* є агрегат, який містить лише компоненти агрегата  $A_1$ , які присутні у агрегаті  $A_2$ , проте він не включає компоненти агрегата  $A_2$ . Наприклад, якщо агрегати  $A_1$  та  $A_2$  є сумісними та мають вигляд:

$$A_1 = \llbracket M_t, M_{hr} | \langle 36.4, 36.1, 36.3, 36.2, 36.5, 36.3 \rangle, \langle 75, 76, 74, 73, 75, 75 \rangle \rrbracket,$$

$$A_2 = \llbracket M_t, M_{hr} | \langle 36.5, 36.5, 36.8, 36.6, 36.3, 36.4, 37.0, 36.5 \rangle, \langle 74, 81, 76, 93, 97, 97, 96 \rangle \rrbracket.$$

Тоді, результатом їх *перерізу* є агрегат:

$$A_1 \cap A_2 = \llbracket M_t, M_{hr} | \langle 36.4, 36.3, 36.5, 36.3, 36.5, 36.5, 36.3, 36.4, 36.5 \rangle, \langle 76, 74, 74, 76 \rangle \rrbracket.$$

Водночас, результатом *виключного перерізу* є агрегат:

$$A_1 \neg A_2 = \llbracket M_t, M_{hr} | \langle 36.4, 36.3, 36.5, 36.3 \rangle, \langle 76, 74 \rangle \rrbracket.$$

Розглянемо агрегат  $A_3$ , який є квазісумісним з агрегатом  $A_1$  та має вигляд:

$$A_3 = \llbracket M_t, M_{sp} | \langle 36.5, 36.5, 36.8, 36.6, 36.3, 36.4, 37.0, 36.5 \rangle, \langle 177, 159, 174, 155, 167, 150, 177, 135 \rangle \rrbracket.$$

Тоді, результатом *перерізу* агрегатів  $A_1$  та  $A_3$  є агрегат:

$$A_1 \cap A_3 = \llbracket M_t | \langle 36.4, 36.3, 36.5, 36.3, 36.5, 36.5, 36.3, 36.4, 36.5 \rangle \rrbracket,$$

а результатом *виключного перерізу* цих агрегатів є агрегат:

$$A_1 \neg A_3 = \llbracket M_t | \langle 36.4, 36.3, 36.5, 36.3 \rangle \rrbracket.$$

Логічні операції в АСА є некомутативними, оскільки порядок слідування компонентів є важливим. Це відрізняє логічні операції над агрегатами від логічних операцій над множинами.

### 2.2.2. Операції впорядкування

Операціями впорядкування є *розміщення, сортування, проріджування, видалення та вставлення* [1, 12].

Операція *розміщення* перевпорядковує кортеж множин та кортеж кортежів агрегата  $A$  відповідно до порядку слідування кортежу множин та кортежу кортежів шаблонного агрегата  $A_{tem}$ . За шаблонний агрегат може бути взятий невизначений, пустий або будь-який довільний агрегат.

Нехай агрегат  $A$  визначений наступним чином:

$$A = \llbracket M_3, M_1, M_2, \dots, M_N | \langle a_{i_3}^3 \rangle_{i_3=1}^{n_3}, \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2}, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket \quad (2.25)$$

та нехай шаблонний агрегат  $A_{tem}$  має вигляд:

$$A_{tem} = \llbracket M_1, M_2, M_3, \dots, M_N | \langle \_ \rangle \rrbracket \quad (2.26)$$

Тоді результатом операції *розміщення* над агрегатом  $A$  за зразком агрегата  $A_{tem}$  є агрегат  $B$  виду:

$$B = A \models A_{tem} = \llbracket \{A_{tem}\} | \langle A \rangle \rrbracket = \llbracket M_1, M_2, M_3, \dots, M_N | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2}, \langle a_{i_3}^3 \rangle_{i_3=1}^{n_3}, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket \quad (2.27)$$

Операція *розміщення* може бути застосована до двох довільних агрегатів  $A_1$  та  $A_2$ , де агрегат  $A_2$  використовується як шаблон. Результат операції *розміщення* залежить від сумісності агрегатів.

Якщо  $A_1 \div A_2$ , то результатом операції *розміщення* є агрегат  $B$ , що дорівнює агрегату  $A_1$ :

$$B = A_1 \models A_2 = \llbracket \{A_2\} | \langle A_1 \rangle \rrbracket = \llbracket \{A_1\} | \langle A_1 \rangle \rrbracket = A_1.$$

Якщо  $A_1 \nmid A_2$  та немає прихованої сумісності агрегатів  $A_1$  та  $A_2$ , то результатом застосування операції *розміщення* є пустий агрегат:

$$B = A_1 \models A_2 = \llbracket \{A_2\} | \langle \emptyset \rangle \rrbracket.$$



Якщо  $A_1 \doteq A_2$  та немає прихованої сумісності агрегатів  $A_1$  та  $A_2$ , то результатом застосування операції *розміщення* є агрегат  $B$  виду:

$$B = A_1 \models A_2 = \left[ \left[ M_j^2 | \langle a_i^j \rangle_{i=1}^{n_j} \right] \right]_{j=1}^N, \quad (2.28)$$

де  $\langle a_i^j \rangle_{i=1}^{n_j} \in M_j^2$ ,  $M_j^2 \subset \{A_2\}$ ,  $\langle a_i^j \rangle_{i=1}^{n_j} \subset \langle A_1 \rangle$ ,  $\langle B \rangle \not\equiv \langle A_1 \rangle$ .

Якщо  $A_1 (\div) A_2$ , то результатом застосування операції *розміщення* є агрегат  $B$  виду:

$$B = A_1 \models A_2 = \left[ \left[ \{A_2\} | \langle a_i^j \rangle_{i=1}^{n_j} \right] \right]_{j=1}^N, \quad (2.29)$$

де  $\langle a_i^j \rangle_{i=1}^{n_j} \in \langle A_1 \rangle$ ,  $\langle B \rangle \equiv \langle A_1 \rangle$ .

Теорема 2.1 (теорема про сумісність).

Якщо  $\widehat{A}_1 = A_1 \models A_2$  та  $\widehat{A}_2 = A_2 \models A_1$ , то  $\widehat{A}_1 \div A_2$  та  $\widehat{A}_2 \div A_1$  для  $\forall A_1, \forall A_2$ , таких, що  $A_1 (\div) A_2$ .

Доведення:

Нехай дані довільні агрегати  $A_1$  та  $A_2$ , такі, що  $A_1 (\div) A_2$ . Застосуємо до них операцію *розміщення* двома способами:

1.  $\widehat{A}_1 = A_1 \models A_2$ ;
2.  $\widehat{A}_2 = A_2 \models A_1$ .

Розглянемо агрегати  $\widehat{A}_1$  та  $A_2$ .

Оскільки агрегат  $\widehat{A}_1$  є результатом застосування операції *розміщення*, то відповідно до (2.29) результатом цієї операції над приховано сумісними

агрегатами є агрегат виду  $\widehat{A}_1 = \llbracket \{A_2\} | \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N$ , де  $\langle a_i^j \rangle_{i=1}^{n_j} \in \langle A_1 \rangle$ ,  $\langle \widehat{A}_1 \rangle \equiv \langle A_1 \rangle$ , тобто  $\widehat{A}_1 = \llbracket \{A_2\} | \langle A_1 \rangle \rrbracket$ .

За визначенням агрегата (Визначення 2.1), агрегат  $A_2$  може бути представлений у вигляді  $A_2 = \llbracket \{A_2\} | \langle A_2 \rangle \rrbracket$ .

За визначенням приховано сумісних агрегатів (Визначення 2.8) два агрегати є приховано сумісними, якщо  $\{A_1\} \not\equiv \{A_2\}$ ,  $|A_1| = |A_2| = N$  та  $\forall M_j \subset \{A_k\}, j = [1, \dots, N], k = [1, 2]$ .

Отже,  $\widehat{A}_1 = \llbracket \{A_2\} | \langle A_1 \rangle \rrbracket$ ,  $A_2 = \llbracket \{A_2\} | \langle A_2 \rangle \rrbracket$ ,  $|\widehat{A}_1| = |A_2| = N$ . Тоді, згідно з визначенням сумісних агрегатів (Визначення 2.5) агрегати  $\widehat{A}_1$  та  $A_2$  є сумісними, оскільки  $|\widehat{A}_1| = |A_2|$  та  $\{\widehat{A}_1\} \equiv \{A_2\}$ . Що і потрібно було довести.

Сумісність агрегатів  $A_1$  та  $\widehat{A}_2$  доводиться аналогічно. ■

Наслідок з теореми 2.1. Практичне значення застосування операції *розміщення* над деякими приховано сумісними агрегатами  $A_1$  та  $A_2$ , полягає у тому, що компоненти одного агрегата ( $A_1$ ) може бути перевпорядкований згідно з порядком слідування компонентів другого агрегата ( $A_2$ ), отже, в результаті застосування операції *розміщення* приховано сумісні агрегати стають сумісними.

Розглянемо агрегати  $A_1$ ,  $A_2$  та  $A_3$ , такі що  $A_1 \doteq A_2$ ,  $A_2 (\div) A_3$  та:

$$A_1 = \llbracket M_t, M_{hr}, M_{sp} | \langle 36.4, 36.1, 36.3, 36.2, 36.5, 36.3 \rangle, \langle 75, 76, 74, 73, 75, 75 \rangle, \langle 163, 161, 164, 165, 168 \rangle \rrbracket,$$

$$A_2 = \llbracket M_t, M_{sp} | \langle 36.5, 36.5, 36.8, 36.6, 36.3, 36.4, 37.0, 36.5 \rangle, \langle 171, 183, 175, 183, 167, 167, 176 \rangle \rrbracket$$

$$A_3 = \llbracket M_{sp}, M_t | \langle 177, 159, 174, 155, 167, 150, 177, 135 \rangle, \langle 37.5, 37.2, 36.8, 37.0, 36.6 \rangle \rrbracket.$$

Тоді в результаті застосування операції *розміщення* отримуємо:

$$A_1 \models A_2 = \llbracket M_t, M_{sp} \mid \langle 36.4, 36.1, 36.3, 36.2, 36.5, 36.3 \rangle, \langle 163, 161, 164, 165, 168 \rangle \rrbracket,$$

$$A_3 \models A_2 = \llbracket M_t, M_{sp} \mid \langle 37.5, 37.2, 36.8, 37.0, 36.6 \rangle, \langle 177, 159, 174, 155, 167, 150, 177, 135 \rangle \rrbracket.$$

Операція *сортування* (*сортування за зростанням* та *сортування за спаданням*) дозволяє отримати новий – відсортований – порядок слідування елементів (за зростанням та за спаданням відповідно) певного кортежу, який будемо називати *первісним кортежем*. Результатом застосування операції *сортування* до агрегата є агрегат, у всіх кортежах якого елементи перепорядковані відповідно до нового порядку слідування індексів елементів відсортованого первісного кортежу.

Нехай  $A = \llbracket M_j \mid \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N$  та  $\exists k$  таке, що  $1 < k < N, k \neq 2$  та  $n_1 > n_k > n_N, n_2 = n_k$ . Тоді результатом застосування операції *сортування за зростанням* агрегата  $A$  відносно елементів кортежу  $\bar{a}^k$  є агрегат  $B$  такий, що:

$$B = A \uparrow \bar{a}^k = \llbracket M_1, M_2, \dots, M_k, \dots, M_N \mid, \langle a_\alpha^1, a_\beta^1, \dots, a_\nu^1, \dots, a_\omega^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\alpha^2, a_\beta^2, \dots, a_\nu^2, \dots, a_\omega^2 \rangle, \dots, \langle a_\alpha^k, a_\beta^k, \dots, a_\nu^k, \dots, a_\omega^k \rangle, \dots, \langle a_\alpha^N, a_\beta^N, \dots, a_\nu^N \rangle \rrbracket, \quad (2.30)$$

де  $a_\alpha^k \leq a_\beta^k \leq \dots \leq a_\nu^k \leq \dots \leq a_\omega^k, a_m^j \in \langle a_i^j \rangle_{i=1}^{n_j}, j = 1 \dots N,$

$m \in [\alpha, \beta, \dots, \nu, \dots, \omega], 1 \leq m \leq n$  та  $n = n_k$ , якщо  $n_j \geq n_k$ , або  $n = n_j$ ,

якщо  $n_j < n_k$ .

Результатом застосування операції *сортування за спаданням* агрегата  $A$  відносно елементів кортежу  $\bar{a}^k$  є агрегат  $B$  такий, що:

$$B = A \downarrow \bar{a}^k = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\omega^1, \dots, a_\nu^1, \dots, a_\beta^1, a_\alpha^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\omega^2, \dots, a_\nu^2, \dots, a_\beta^2, a_\alpha^2 \rangle, \dots, \langle a_\omega^k, \dots, a_\nu^k, \dots, a_\beta^k, a_\alpha^k \rangle, \dots, \langle a_\nu^N, \dots, a_\beta^N, a_\alpha^N \rangle \rrbracket. \quad (2.31)$$

де  $a_\alpha^k \geq a_\beta^k \geq \dots \geq a_\nu^k \geq \dots \geq a_\omega^k$ ,  $a_m^j \in \langle a_i^j \rangle_{i=1}^{n_j}$ ,  $j = 1 \dots N$ ,

$m \in [\alpha, \beta, \dots, \nu, \dots, \omega]$ ,  $1 \leq m \leq n$  та  $n = n_k$ , якщо  $n_j \geq n_k$ , або  $n = n_j$ ,

якщо  $n_j < n_k$ .

Якщо  $k = 1, k = 2$  або  $k = N$ , то операція сортування виконується аналогічно.

Якщо  $n_1 = n_2 = \dots = n_k = \dots = n_N$ , то результатом сортування за зростанням є агрегат виду:

$$B = A \uparrow \bar{a}^k = \llbracket M_j | \langle a_\alpha^j, a_\beta^j, \dots, a_\nu^j, \dots, a_\omega^j \rangle \rrbracket_{j=1}^N, \quad (2.32)$$

а результатом сортування за спаданням є агрегат виду:

$$B = A \downarrow \bar{a}^k = \llbracket M_j | \langle a_\omega^j, \dots, a_\nu^j, \dots, a_\beta^j, a_\alpha^j \rangle \rrbracket_{j=1}^N \quad (2.33)$$

Різновидом операцій сортування є операції сортування з доповненням (сортування за зростанням з доповненням та сортування за спаданням з доповненням), які дозволяють збільшити довжину коротших кортежів відповідно до довжини первісного кортежу шляхом додавання значення  $x$  ( $x \in [\emptyset, \_ q]$ ,  $q \in M_j$ ,  $1 \leq j \leq N$ ) або у кінець, або у початок коротших кортежів.

Результатом сортування за зростанням з доповненням у кінець коротших кортежів агрегата  $A$  є агрегат  $B$ :

$$B = A \uparrow (\bar{a}^k, x) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N |, \langle a_\alpha^1, a_\beta^1, \dots, a_\nu^1, \dots, a_\omega^1, \\ a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\alpha^2, a_\beta^2, \dots, a_\nu^2, \dots, a_\omega^2 \rangle, \dots, \\ \langle a_\alpha^k, a_\beta^k, \dots, a_\nu^k, \dots, a_\omega^k \rangle, \dots, \langle a_\alpha^N, a_\beta^N, \dots, a_\nu^N, x, \dots, x \rangle \rrbracket \quad (2.34)$$

де  $|\langle a_\alpha^N, a_\beta^N, \dots, a_\nu^N, x, \dots, x \rangle| = n_k$ .

Результатом сортування за зростанням з доповненням напочатку коротших кортежів агрегата  $A$  є агрегат  $B$ :

$$B = A \uparrow (x, \bar{a}^k) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N |, \langle a_\alpha^1, a_\beta^1, \dots, a_\nu^1, \dots, a_\omega^1, \\ a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\alpha^2, a_\beta^2, \dots, a_\nu^2, \dots, a_\omega^2 \rangle, \dots, \\ \langle a_\alpha^k, a_\beta^k, \dots, a_\nu^k, \dots, a_\omega^k \rangle, \dots, \langle x, \dots, x, a_\alpha^N, a_\beta^N, \dots, a_\nu^N \rangle \rrbracket, \quad (2.35)$$

де  $|\langle x, \dots, x, a_\alpha^N, a_\beta^N, \dots, a_\nu^N \rangle| = n_k$ .

Результатом сортування за спаданням з доповненням у кінець коротших кортежів агрегата  $A$  є агрегат  $B$ :

$$B = A \downarrow (\bar{a}^k, x) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N |, \langle a_\omega^1, \dots, a_\nu^1, \dots, a_\beta^1, \\ a_\alpha^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\omega^2, \dots, a_\nu^2, \dots, a_\beta^2, a_\alpha^2 \rangle, \dots, \\ \langle a_\omega^k, \dots, a_\nu^k, \dots, a_\beta^k, a_\alpha^k \rangle, \dots, \langle a_\nu^N, \dots, a_\beta^N, a_\alpha^N, x, \dots, x \rangle \rrbracket. \quad (2.36)$$

Результатом сортування за спаданням з доповненням напочатку коротших кортежів агрегата  $A$  є агрегат  $B$ :

$$B = A \downarrow (x, \bar{a}^k) = \llbracket M_1, M_2, \dots, M_k, \dots, M_N |, \langle a_\omega^1, \dots, a_\nu^1, \dots, a_\beta^1, \\ a_\alpha^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\omega^2, \dots, a_\nu^2, \dots, a_\beta^2, a_\alpha^2 \rangle, \dots, \\ \langle a_\omega^k, \dots, a_\nu^k, \dots, a_\beta^k, a_\alpha^k \rangle, \dots, \langle x, \dots, x, a_\nu^N, \dots, a_\beta^N, a_\alpha^N \rangle \rrbracket. \quad (2.37)$$

Операція *проріджування* дозволяє вилучити з первісного кортежу агрегата однакові значення, що розташовані поруч; одночасно вилучаються елементи з тими самими індексами у всіх інших кортежах. Якщо агрегат  $A$  визначений як:

$$A = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_1^1, \dots, a_m^1, \dots, a_{m+p}^1, a_{m+p+1}^1, \dots, a_{n_1}^1 \rangle, \dots, \langle a_1^k, \dots, a_m^k, \dots, a_{m+p}^k, a_{m+p+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_1^N, \dots, a_m^N, \dots, a_{m+p}^N, a_{m+p+1}^N, \dots, a_{n_N}^N \rangle \rrbracket, \quad (2.38)$$

де  $1 \leq k \leq N$ , та  $\exists m_l, \forall l$ , таке, що  $a_m^k = a_{m+1}^k = \dots = a_{m+p}^k$ ,  $1 \leq m \leq (n_k - p)$ ,  $1 \leq p \leq n_k$ , то результатом операції проріджування агрегата  $A$  відносно кортежу  $\bar{a}^k$  є агрегат  $B$  виду:

$$B = A \parallel \bar{a}^k = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_1^1, \dots, a_m^1, a_{m+p+1}^1, \dots, a_{n_1}^1 \rangle, \dots, \langle a_1^k, \dots, a_m^k, a_{m+p+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_1^N, \dots, a_m^N, a_{m+p+1}^N, \dots, a_{n_N}^N \rangle \rrbracket. \quad (2.39)$$

Операція *видалення* дозволяє вилучити певний елемент з агрегата. Нехай агрегат  $A$  має вигляд:

$$A = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, a_m^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket, \quad (2.40)$$

тоді результатом операції *видалення* елементу  $a_m^k$  з агрегата  $A$  є агрегат  $B$  такий, що:

$$B = A \bowtie a_m^k = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket \quad (2.41)$$

Різновидом операції видалення є операція *умовного видалення*, яка передбачає видалення елемента в разі виконання певної умови, як-то  $a_m^k = d$ ,  $a_m^k < d$  або  $a_m^k > d$ ,  $\forall d \in M_k$ . Наприклад:  $B = A \bowtie a_m^k \big|_{a_m^k=d}$ .

Операція *вставлення* дозволяє додати певний елемент у агрегат. Нехай агрегат  $A$  має вигляд:

$$A = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, a_m^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket, \quad (2.42)$$

де  $1 \leq m \leq N$  та  $\exists d$ , таке що  $d \in M_k$ ,  $1 \leq k \leq N$ , то вставлення елемента  $d$  в одне і те саме місце у агрегаті  $A$  може бути виконане двома еквівалентними способами, що визначені у (2.43) та (2.44).

$$B = A \bowtie (d < a_m^k) = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, d, a_m^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket \quad (2.43)$$

$$B = A \bowtie (d > a_{m-1}^k) = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}, \dots, \langle a_1^k, \dots, a_{m-1}^k, d, a_m^k, a_{m+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_{i_N}^N \rangle_{i_N=1}^{n_N} \rrbracket \quad (2.44)$$

Різновидом операції видалення є операція *умовного вставлення*, яка передбачає додавання елемента в разі виконання певної умови:  $d = a_m^k$ ,  $d < a_m^k$  або  $d > a_m^k$ ,  $\forall d \in M_k$ .

Наприклад,  $B = A \bowtie (d < a_m^k) \big|_{d=a_m^k, m=m_1 \dots m_2}$ .

Отже, операції впорядкування, визначені у АСА, дозволяють перевпорядковувати елементи у кортежах та кортежі у агрегатах.

### 2.2.3. Арифметичні операції

Арифметичні операції в АСА можуть виконуватись лише над елементами, які належать до числових множин.

Арифметичними операціями над елементами є додавання (+), віднімання (−), множення (\*), ділення (/). Вони виконуються над окремими значеннями за звичайними правилами виконання цих математичних операцій.

Арифметичними операціями над кортежами є поелементне додавання (+.), поелементне віднімання (−.), поелементне множення (\*.), поелементне ділення (/.). Ці операції виконуються за такими правилами:

1. Відповідність між елементами різних кортежів встановлюється зліва направо (від першого елемента до останнього).
2. Якщо кортежі, над якими виконується поелементна операція, мають різну довжину, то операція виконується лише для  $k$  перших елементів кожного кортежу-операнду, де  $k$  – довжина найкоротшого кортежу-операнду; інші елементи всіх довших кортежів-операндів відкидаються.

Наприклад, результатом додавання двох кортежів дійсних чисел:  $\langle 2.5, 3.1, 7.4 \rangle$  та  $\langle 0.3, 12.8, 1.4, 6.9, 3.2 \rangle$  є кортеж  $\langle 2.8, 15.9, 8.8 \rangle$ .

### 2.3. Відношення в алгебраїчній системі агрегатів

Відношення в АСА [1, 3] включають відношення між елементами, відношення між кортежами та відношення між агрегатами.

Відношеннями між елементами є відношення *більше* ( $>$ ), *менше* ( $<$ ), *дорівнює* ( $=$ ), *передуює* ( $<$ ), *слідуює за* ( $>$ ). Перші три відношення ( $<$ ,  $>$  та  $=$ )



ґрунтуються на порівнянні значень елементів, в той час, як два останніх відношення ( $<$  та  $>$ ) ґрунтуються на порівнянні позиції елементів у кортежі.

Розглянемо елементи такого кортежу:

$$\bar{a} = \langle a_1, a_2, a_3, a_4 \rangle = \langle 11, 9, 11, 18 \rangle.$$

Тоді між елементами цього кортежу можна встановити наступні відношення:  $a_1 > a_2$ ;  $a_3 < a_4$ ;  $a_1 = a_3$ ;  $a_1 < a_2$ ;  $a_3 > a_2$ .

### 2.3.1. Відношення між кортежами

Відношенням між кортежами належать до таких категорій:

- арифметичні відношення;
- частотні відношення;
- інтервальні відношення.

*Арифметичні відношення* можуть бути застосовані до двох кортежів  $\bar{a}^1$  та  $\bar{a}^2$ , де  $\bar{a}^1 = \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}$  та  $\bar{a}^2 = \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2}$ , якщо  $a_{i_1}^1 \in M$  та  $a_{i_2}^2 \in M$ . Арифметичні відношення виконуються поелементно та включають наступні відношення: *стро́го більше* ( $>$ ), *мажоритарно більше* ( $>>$ ), *стро́го менше* ( $<$ ), *мажоритарно менше* ( $<<$ ), *стро́го дорівнює* ( $=$ ) та *мажоритарно дорівнює* ( $<>$ ).

Відношення *стро́го більше* між кортежами  $\bar{a}^1$  та  $\bar{a}^2$  визначається як:

$$\bar{a}^1 > \bar{a}^2 \text{ якщо } a_i^1 > a_i^2, \forall i, i = [1..n_1], n_1 = n_2. \quad (2.45)$$

Відношення *мажоритарно більше* між кортежами  $\bar{a}^1$  та  $\bar{a}^2$  може бути визначене наступним чином. Нехай  $N = \langle 1, 2, \dots, n \rangle$ , де  $n = \begin{cases} n_1, & \text{if } n_1 \leq n_2 \\ n_2, & \text{if } n_1 > n_2 \end{cases}$ ,

та  $\exists N_p \neq \emptyset, \exists N_q \neq \emptyset$  такі, що  $N_p \cup N_q = N, N_p \cap N_q = \emptyset$  та  $|N_p| > |N_q|$ .  
Тоді  $\forall p, \forall q, p \in N_p, q \in N_q$  відношення *мажоритарно більше*:

$$\overline{a^1} \gg \overline{a^2}, \text{ якщо } a_p^1 > a_p^2, a_q^1 \leq a_q^2. \quad (2.46)$$

Відношення *строго менше* між кортежами  $\overline{a^1}$  та  $\overline{a^2}$  визначається як:

$$\overline{a^1} < \overline{a^2}, \text{ якщо } a_i^1 < a_i^2, \forall i, i = [1..n_1], n_1 = n_2. \quad (2.47)$$

Відношення *мажоритарно менше* між кортежами  $\overline{a^1}$  та  $\overline{a^2}$  може бути визначене наступним чином. Нехай  $N = \langle 1, 2, \dots, n \rangle$ , де  $n = \begin{cases} n_1, \text{ if } n_1 \leq n_2 \\ n_2, \text{ if } n_1 > n_2 \end{cases}$  та  $\exists N_p \neq \emptyset, \exists N_q \neq \emptyset$  такі, що  $N_p \cup N_q = N, N_p \cap N_q = \emptyset$  та  $|N_p| < |N_q|$ .  
Тоді  $\forall p, \forall q, p \in N_p, q \in N_q$ , відношення *мажоритарно менше*:

$$\overline{a^1} \ll \overline{a^2}, \text{ якщо } a_p^1 \geq a_p^2, a_q^1 < a_q^2. \quad (2.48)$$

Відношення *строго дорівнює* між кортежами  $\overline{a^1}$  та  $\overline{a^2}$  визначається наступним чином:

$$\overline{a^1} = \overline{a^2}, \text{ якщо } a_i^1 = a_i^2, \forall i, i = [1..n_1], n_1 = n_2. \quad (2.49)$$

Відношення *мажоритарно дорівнює* між кортежами  $\overline{a^1}$  та  $\overline{a^2}$  може бути визначене наступним чином.

Нехай  $N = \langle 1, 2, \dots, n \rangle$ , де  $n = \begin{cases} n_1, \text{ if } n_1 \leq n_2 \\ n_2, \text{ if } n_1 > n_2 \end{cases}$  та  $\exists N_e \neq \emptyset, \exists N_p \neq \emptyset, \exists N_q \neq \emptyset$  такі, що  $N_p \cup N_q \cup N_e = N, N_p \cap N_q \cap N_e = \emptyset$  та

$|N_e| > |N_p|$ ,  $|N_p| = |N_q|$ . Тоді  $\forall p \in N_p, \forall q \in N_q, \forall e \in N_e$  відношення мажоритарно дорівнює:

$$\overline{a^1} <> \overline{a^2}, \text{ якщо } a_p^1 > a_p^2, a_q^1 < a_q^2, a_e^1 = a_e^2 \quad (2.50)$$

Розглянемо наступні кортежі:

$$\overline{a^1} = \langle a_1^1, a_2^1, a_3^1, a_4^1 \rangle = \langle 11, 9, 11, 18 \rangle;$$

$$\overline{a^2} = \langle a_1^2, a_2^2, a_3^2, a_4^2 \rangle = \langle 2, 7, 4, 10 \rangle;$$

$$\overline{a^3} = \langle a_1^3, a_2^3, a_3^3, a_4^3, a_5^3 \rangle = \langle 7, 19, 4, 10, 8 \rangle;$$

$$\overline{a^4} = \langle a_1^4, a_2^4, a_3^4, a_4^4 \rangle = \langle 11, 9, 11, 18 \rangle;$$

$$\overline{a^5} = \langle a_1^5, a_2^5, a_3^5, a_4^5, a_5^5 \rangle = \langle 14, 9, 11, 10, 8 \rangle.$$

Тоді між цими кортежами можна встановити наступні відношення:

$$\begin{aligned} \overline{a^1} > \overline{a^2}; \quad \overline{a^1} \gg \overline{a^3}; \quad \overline{a^2} < \overline{a^4}; \\ \overline{a^3} \ll \overline{a^4}; \quad \overline{a^1} = \overline{a^4}; \quad \overline{a^1} <> \overline{a^5}. \end{aligned}$$

Частотні відношення можуть бути застосовані до двох кортежів  $\overline{a^1}$  та  $\overline{a^2}$ , де  $\overline{a^1} = \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1}$  та  $\overline{a^2} = \langle a_{i_2}^2 \rangle_{i_2=1}^{n_2}$ , якщо  $a_{i_1}^1 \in M$  та  $a_{i_2}^2 \in M$ . Частотні відношення включають наступні відношення: *частіше* ( $\triangleright$ ), *рідше* ( $\triangleleft$ ) та *однаково часто* ( $\sim$ ).

Відношення *частіше* між кортежами  $\overline{a^1}$  та  $\overline{a^2}$  визначається як:

$$\overline{a^1} \triangleright \overline{a^2}, \text{ якщо } |\overline{a^1}| > |\overline{a^2}|. \quad (2.51)$$

Відношення *рідше* між кортежами  $\overline{a^1}$  та  $\overline{a^2}$  визначається як:

$$\overline{a^1} \triangleleft \overline{a^2}, \text{ якщо } |\overline{a^1}| < |\overline{a^2}|. \quad (2.52)$$

Відношення *однаково часто* між кортежами  $\overline{a^1}$  та  $\overline{a^2}$  визначається як:

$$\overline{a^1} \sim \overline{a^2}, \text{ якщо } |\overline{a^1}| = |\overline{a^2}|. \quad (2.53)$$

Розглянемо наступні кортежі:

$$\overline{a^1} = \langle a_1^1, a_2^1, a_3^1, a_4^1, a_5^1 \rangle = \langle 14, 9, 15, 18, 6 \rangle;$$

$$\overline{a^2} = \langle a_1^2, a_2^2, a_3^2, a_4^2 \rangle = \langle 2, 7, 4, 10 \rangle;$$

$$\overline{a^3} = \langle a_1^3, a_2^3, a_3^3, a_4^3, a_5^3 \rangle = \langle 7, 19, 4, 10, 8 \rangle;$$

Тоді, між ними можна встановити такі відношення:  $\overline{a^1} \triangleright \overline{a^2}$ ;  $\overline{a^2} \triangleleft \overline{a^3}$ ;  $\overline{a^1} \sim \overline{a^3}$ .

Для того, щоб визначити *наскільки частішим* або *наскільки рідшим* є певний кортеж по відношенню до іншого, введемо *міру частоти*:

$$\eta = \frac{|\overline{a^1}|}{|\overline{a^2}|} \quad (2.54)$$

Наприклад, для кортежів  $\overline{a^1}$ ,  $\overline{a^2}$  та  $\overline{a^3}$ , визначених вище, ця міра має значення:  $\eta_{12} = 1.25, \eta_{23} = 0.8, \eta_{13} = 1$ .

*Інтервальні відношення* включають наступні відношення: *передую*, *настає після*, *збігається з*, *стикається з початком*, *стикається з кінцем*, *перекриває*, *перекривається*, *відбувається під час*, *містить*, *починає*, *починається з*, *закінчує*, *закінчується* та *відбувається між*.

Інтервальні відношення можуть бути застосовані лише до кортежів, які є дискретними інтервалами. Розглянемо поняття дискретного інтервалу докладно.

### 2.3.2. Відношення над дискретними інтервалами

Інтервальні відношення ґрунтуються на відношеннях інтервальної алгебри Аллена [265–274]. Якщо  $X$  та  $Y$  є інтервалами, такими що  $X = [x^-, x^+]$  та  $Y = [y^-, y^+]$ , відношення між ними визначаються, як наведено у табл. 2.1.

Суттєва відмінність інтервальних відношень в АСА від інтервальної алгебри Аллена полягає у тому, що, на відміну від інтервальної алгебри Аллена, яка оперує неперервними інтервалами, інтервальні відношення в АСА застосовуються до дискретних інтервалів або дискретних інтервалів без повторень.

Таблиця 2.1 – Відношення інтервальної алгебри Аллена

Відношення	Запис	Визначення
$X \text{ Before } Y$	$b(X, Y)$	$x^+ < y^-$
$X \text{ After } Y$	$bi(X, Y)$	$x^- > y^+$
$X \text{ Equal } Y$	$e(X, Y)$	$x^- = y^-$ та $x^+ = y^+$
$X \text{ Meets } Y$	$m(X, Y)$	$x^+ = y^-$
$X \text{ Is Met By } Y$	$mi(X, Y)$	$x^- = y^+$
$X \text{ Overlaps } Y$	$o(X, Y)$	$x^- < y^-$ та $x^+ > y^-$ та $x^+ < y^+$
$X \text{ Is Overlapped By } Y$	$oi(X, Y)$	$x^- > y^-$ та $x^- < y^+$ та $x^+ > y^+$
$X \text{ During } Y$	$d(X, Y)$	$x^- > y^-$ та $x^+ < y^-$
$X \text{ Contains } Y$	$di(X, Y)$	$x^- < y^-$ та $x^- > y^+$
$X \text{ Starts } Y$	$s(X, Y)$	$x^- = y^-$ та $x^+ < y^+$
$X \text{ Is Started By } Y$	$si(X, Y)$	$x^- = y^-$ та $x^+ > y^+$
$X \text{ Finishes } Y$	$f(X, Y)$	$x^- > y^-$ та $x^+ = y^+$
$X \text{ Is Finished By } Y$	$fi(X, Y)$	$x^- < y^-$ та $x^+ = y^+$

Визначення 2.9. *Дискретним інтервалом* є кортеж  $\bar{a} = \langle a_i \rangle_{i=1}^n$ , елементами якого можуть бути будь-які значення  $a_i$ , впорядковані за зростанням ( $a_i \leq a_{i+1}, \forall i \in [1 \dots n-1], a_i \in M$ ) або за спаданням ( $a_i \geq a_{i+1}, \forall i \in [1 \dots n-1], a_i \in M$ ), тобто дискретний інтервал є зростаючою або спадною послідовністю, що визначена на деякій множині  $M$ .

Визначення 2.10. *Дискретним інтервалом без повторень* будемо називати кортеж  $\bar{a} = \langle a_i \rangle_{i=1}^n$ , елементи якого є унікальними значеннями  $a_i$ , впорядкованими за зростанням ( $a_i < a_{i+1}, \forall i \in [1 \dots n-1], a_i \in M$ ) або за спаданням ( $a_i > a_{i+1}, \forall i \in [1 \dots n-1], a_i \in M$ ), тобто дискретний інтервал без повторень є строго зростаючою або строго спадною послідовністю, що визначена на деякій множині  $M$ .

Теорема 2.2 (теорема про сортування).

Якщо  $\bar{a}^k \in A$  ( $\forall k \in [1 \dots N], N \in \mathbb{N}, \forall A = \llbracket M_j | \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N$ ) є первісним кортежем операцій сортування  $\hat{A} = A \uparrow \bar{a}^k$  та  $\hat{A} = A \downarrow \bar{a}^k$ , то  $\widehat{\bar{a}^k} = \langle \widehat{a_i^k} \rangle_{i=1}^{n_k} \in \hat{A} = \llbracket M_j | \langle \hat{a_i^j} \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N$  є дискретним інтервалом.

Доведення:

Нехай даний довільний агрегат  $A = \llbracket M_j | \langle a_i^j \rangle_{i=1}^{n_j} \rrbracket_{j=1}^N$ . Застосуємо до нього операцію сортування за зростанням відносно деякого первісного кортежу  $\bar{a}^k$  ( $k \in [1 \dots N], N \in \mathbb{N}$ ). Згідно з (2.30) в результаті отримуємо агрегат  $\hat{A}$  виду:

$$\hat{A} = A \uparrow \bar{a}^k = \llbracket M_1, M_2, \dots, M_k, \dots, M_N | \langle a_\alpha^1, a_\beta^1, \dots, a_\nu^1, \dots, a_\omega^1, a_{n_k+1}^1, \dots, a_{n_1}^1 \rangle, \langle a_\alpha^2, a_\beta^2, \dots, a_\nu^2, \dots, a_\omega^2 \rangle, \dots, \langle a_\alpha^k, a_\beta^k, \dots, a_\nu^k, \dots, a_\omega^k \rangle, \dots, \langle a_\alpha^N, a_\beta^N, \dots, a_\nu^N \rangle \rrbracket,$$

такий що  $a_\alpha^k < a_\beta^k < \dots < a_\nu^k < \dots < a_\omega^k$ ,  $a_m^j \in \langle a_i^j \rangle_{i=1}^{n_j}$ ,  $j = 1 \dots N$ ,  
 $m \in [\alpha, \beta, \dots, \nu, \dots, \omega]$ ,  $1 \leq m \leq n$  та  $n = n_k$ , якщо  $n_j \geq n_k$ , або  $n = n_j$ ,  
якщо  $n_j < n_k$ .

Оскільки порядок слідування елементів кортежів агрегата  $A$  в результаті застосування операції *сортування за зростанням* змінено згідно з новим порядком слідування елементів кортежу  $\hat{A}$ , то позначимо ці перевпорядковані кортежі як  $\langle \hat{a}_i^j \rangle_{i=1}^{n_j}$ . Тоді  $\hat{A} = A \uparrow \bar{a}^k = \left[ \left[ M_j | \langle \hat{a}_i^j \rangle_{i=1}^{n_j} \right]_{j=1}^N \right]$ .

У цьому відсортованому агрегаті елементи первісного кортежу  $\widehat{\bar{a}}^k = \left\langle \widehat{a}_i^k \right\rangle_{i=1}^{n_k}$  є відсортованими за зростанням згідно з (2.30).

Отже, кортеж  $\widehat{\bar{a}}^k$  є дискретним інтервалом, оскільки за визначенням (Визначення 2.9) дискретний інтервал – це кортеж, елементи якого впорядковані за зростанням ( $a_i \leq a_{i+1}$ ,  $\forall i \in [1 \dots n - 1]$ ,  $a_i \in M$ ). Що і потрібно було довести.

Доведення для випадку операції *сортування за спаданням* є аналогічним. ■

### Теорема 2.3 (теорема про проріджування).

Якщо первісний кортеж  $\bar{a}^k \in A$  операції проріджування  $\hat{A} = A \parallel \bar{a}^k$  ( $\forall k \in [1 \dots N]$ ,  $N \in \mathbb{N}$ ,  $\forall A = \left[ \left[ M_j | \langle a_i^j \rangle_{i=1}^{n_j} \right]_{j=1}^N \right]$ ) є дискретним інтервалом, то  $\widehat{\bar{a}}^k = \left\langle \widehat{a}_i^k \right\rangle_{i=1}^{n_k} \in \hat{A} = \left[ \left[ M_j | \langle \hat{a}_i^j \rangle_{i=1}^{n_j} \right]_{j=1}^N \right]$  є дискретним інтервалом без повторень.

Доведення:

Нехай даний довільний агрегат  $A = \left[ \left[ M_j | \langle a_i^j \rangle_{i=1}^{n_j} \right]_{j=1}^N \right]$ , деякий кортеж  $\bar{a}^k$  якого є дискретним інтервалом ( $\forall k \in [1 \dots N]$ ,  $N \in \mathbb{N}$ ).

За визначенням дискретного інтервалу (Визначення 2.9) дискретним інтервалом є довільний кортеж, елементами якого можуть бути будь-які значення, впорядковані за зростанням або за спаданням. Нехай виконується нерівність:

$$a_i^k \leq a_{i+1}^k \quad (\forall i \in [1 \dots n - 1], a_i \in M). \quad (2.55)$$

Застосуємо до агрегата  $A$  операцію *проріджування* відносно  $\bar{a}^k$  ( $\forall k \in [1 \dots N]$ ,  $N \in \mathbb{N}$ ). У випадках, коли нерівність (2.55) набуває вигляду  $a_i^k = a_{i+1}^k$  ( $\forall i \in [1 \dots n - 1]$ ), й оскільки за визначенням (2.39) операція *проріджування* вилучає з первісного кортежу агрегата однакові значення, що розташовані поруч, то згідно з (2.39) в результаті операції *проріджування* отримуємо агрегат  $\hat{A}$  виду:

$$\hat{A} = A \parallel \bar{a}^k = \llbracket M_1, \dots, M_k, \dots, M_N | \langle a_1^1, \dots, a_m^1, a_{m+p+1}^1, \dots, a_{n_1}^1 \rangle, \dots, \langle a_1^k, \dots, a_m^k, a_{m+p+1}^k, \dots, a_{n_k}^k \rangle, \dots, \langle a_1^N, \dots, a_m^N, a_{m+p+1}^N, \dots, a_{n_N}^N \rangle \rrbracket,$$

для якого нерівність (2.55) набуває вигляду  $a_i^k < a_{i+1}^k$ .

Оскільки кількість елементів кортежів агрегата  $A$  в результаті застосування операції *проріджування* у загальному випадку зменшується, то позначимо ці проріджені кортежі як  $\langle \hat{a}_i^j \rangle_{i=1}^{\hat{n}_j}$  ( $\hat{n}_j \leq n_j$ ). Тоді  $\hat{A} = A \parallel \bar{a}^k = \llbracket M_j | \langle \hat{a}_i^j \rangle_{i=1}^{\hat{n}_j} \rrbracket_{j=1}^N$ . У цьому прорідженому агрегаті елементи первісного кортежу  $\widehat{\bar{a}}^k = \langle \widehat{a}_i^k \rangle_{i=1}^{n_k}$  підпадають під дію умови  $\widehat{a}_i^k < \widehat{a}_{i+1}^k$ . Отже, кортеж  $\widehat{\bar{a}}^k$  є дискретним інтервалом без повторень, оскільки за визначенням (Визначення 2.10) дискретний інтервал без повторень – це кортеж, елементи якого впорядковані строго за зростанням ( $a_i < a_{i+1}$ ,  $\forall i \in [1 \dots n - 1]$ ,  $a_i \in M$ ). Що і потрібно було довести.



Доведення для випадку, коли умова (2.55) має вигляд  $a_i^k \geq a_{i+1}^k$ , є аналогічним. ■

Розглянемо відношення між дискретними інтервалами. Для узгодженості з інтервальною алгеброю Аллена будемо позначати ці відношення індексом  $d$ , наприклад,  $e_d$ . Тоді відношеннями між дискретними інтервалами  $\overline{a^1}$  та  $\overline{a^2}$  визначимо наступним чином.

Відношення  $\overline{a^1}$  *передуює*  $\overline{a^2}$  визначається як:

$$b_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_{n_1}^1 < a_1^2. \quad (2.56)$$

Відношення  $\overline{a^1}$  *настає після*  $\overline{a^2}$  визначається як:

$$bi_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 > a_{n_2}^2. \quad (2.57)$$

Відношення  $\overline{a^1}$  *збігається з*  $\overline{a^2}$  визначається як:

$$e_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 = a_1^2, a_{n_1}^1 = a_{n_2}^2. \quad (2.58)$$

Відношення  $\overline{a^1}$  *стикається з початком*  $\overline{a^2}$  визначається як:

$$m_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_{n_1}^1 = a_1^2. \quad (2.59)$$

Відношення  $\overline{a^1}$  *стикається з кінцем*  $\overline{a^2}$  визначається як:

$$mi_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 = a_{n_2}^2. \quad (2.60)$$

Відношення  $\overline{a^1}$  *перекриває*  $\overline{a^2}$  визначається як:

$$o_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 < a_1^2 \text{ та } a_{n_1}^1 < a_{n_2}^2 \text{ та } a_1^2 < a_{n_1}^1. \quad (2.61)$$

Відношення  $\overline{a^1}$  *перекривається*  $\overline{a^2}$  визначається як:

$$oi_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^2 < a_1^1 \text{ та } a_{n_2}^2 < a_{n_1}^1 \text{ та } a_1^1 < a_{n_2}^2. \quad (2.62)$$

Відношення  $\overline{a^1}$  *відбувається під час*  $\overline{a^2}$  визначається як:

$$d_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 > a_1^2 \text{ та } a_{n_1}^1 < a_{n_2}^2. \quad (2.63)$$

Відношення  $\overline{a^1}$  *містить*  $\overline{a^2}$  визначається як:

$$di_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 < a_1^2 \text{ та } a_{n_1}^1 > a_{n_2}^2. \quad (2.64)$$

Відношення  $\overline{a^1}$  *починає*  $\overline{a^2}$  визначається як:

$$s_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 = a_1^2 \text{ та } a_{n_1}^1 < a_{n_2}^2. \quad (2.65)$$

Відношення  $\overline{a^1}$  *починається з*  $\overline{a^2}$  визначається як:

$$si_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 = a_1^2 \text{ та } a_{n_1}^1 > a_{n_2}^2. \quad (2.66)$$

Відношення  $\overline{a^1}$  *закінчує*  $\overline{a^2}$  визначається як:

$$f_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 > a_1^2 \text{ та } a_{n_1}^1 = a_{n_2}^2. \quad (2.67)$$

Відношення  $\overline{a^1}$  закінчується  $\overline{a^2}$  визначається як:

$$fi_d(\overline{a^1}, \overline{a^2}), \text{ якщо } a_1^1 < a_1^2 \text{ та } a_{n_1}^1 = a_{n_2}^2. \quad (2.68)$$

Відношення *відбувається між*  $w_d^{(\alpha, \beta)}(\overline{a^1}, \dots, \overline{a^n})$  не має аналогу в інтервальній алгебрі Аллена. Це відношення може бути застосоване до будь-якої кількості дискретних інтервалів та означає, що  $\overline{a^j}$  ( $j \in [1 \dots n]$ ,  $n \geq 1$ ) *відбувається між* дискретними значеннями  $\alpha$  та  $\beta$ . Отже, відношення  $\overline{a^1}$  та  $\overline{a^2}$  *відбуваються між*  $\alpha$  та  $\beta$  визначається як:

$$w_d^{(\alpha, \beta)}(\overline{a^1}, \overline{a^2}), \text{ якщо } \alpha \leq a_1^1, \alpha \leq a_1^2, a_{n_1}^1 \leq \beta, a_{n_2}^2 \leq \beta, \quad (2.69)$$

де  $\alpha$  та  $\beta$  – задані дискретні значення такі, що  $\alpha, \beta, a_i^j \in M$ ,  $j \in [1 \dots n]$ ,  $i \in [1 \dots n_j]$ ,  $n, n_j \in \mathbb{N}$ ;  $M$  – деяка множина.

Відношення *відбувається між* для двох дискретних інтервалів є узагальненням відношення *збігається з*, оскільки відношення *відбувається між* (2.69) є еквівалентом відношенню *збігається з* (2.58), якщо  $a_1^1 = a_1^2$  та  $a_{n_1}^1 = a_{n_2}^2$ .

Розглянемо наступні кортежі (рис. Рис. 2.1. Відношення *передують*-Рис. 2.14. Відношення *відбувається між*):

$$\overline{a^1} = \langle a_1^1, a_2^1, a_3^1 \rangle = \langle 1, 3, 4 \rangle;$$

$$\overline{a^2} = \langle a_1^2, a_2^2, a_3^2, a_4^2, a_5^2 \rangle = \langle 8, 9, 12, 13, 16 \rangle;$$

$$\overline{a^3} = \langle a_1^3, a_2^3, a_3^3, a_4^3 \rangle = \langle 1, 3, 5, 6 \rangle;$$

$$\overline{a^4} = \langle a_1^4, a_2^4, a_3^4, a_4^4 \rangle = \langle 8, 10, 14, 16 \rangle;$$

$$\overline{a^5} = \langle a_1^5, a_2^5, a_3^5, a_4^5, a_5^5, a_6^5 \rangle = \langle 2, 4, 5, 6, 7, 8 \rangle;$$

$$\overline{a^6} = \langle a_1^6, a_2^6, a_3^6 \rangle = \langle 6, 8, 10 \rangle;$$

$$\overline{a^7} = \langle a_1^7, a_2^7, a_3^7, a_4^7, a_5^7 \rangle = \langle 5, 7, 10, 15, 18 \rangle;$$

$$\overline{a^8} = \langle a_1^8, a_2^8, a_3^8, a_4^8 \rangle = \langle 8, 10, 11, 12 \rangle;$$

$$\overline{a^9} = \langle a_1^9, a_2^9, a_3^9, a_4^9, a_5^9 \rangle = \langle 5, 7, 12, 15, 16 \rangle;$$

$$\overline{a^{10}} = \langle a_1^{10}, a_2^{10}, a_3^{10} \rangle = \langle 4, 7, 8 \rangle.$$

Тоді між ними можна встановити наступні відношення:  $b_d(\overline{a^1}, \overline{a^2})$ ,  $bi_d(\overline{a^2}, \overline{a^3})$ ,  $e_d(\overline{a^2}, \overline{a^4})$ ,  $m_d(\overline{a^5}, \overline{a^2})$ ,  $mi_d(\overline{a^4}, \overline{a^5})$ ,  $o_d(\overline{a^3}, \overline{a^5})$ ,  $oi_d(\overline{a^5}, \overline{a^6})$ ,  $d_d(\overline{a^6}, \overline{a^7})$ ,  $di_d(\overline{a^7}, \overline{a^2})$ ,  $s_d(\overline{a^1}, \overline{a^3})$ ,  $si_d(\overline{a^2}, \overline{a^8})$ ,  $f_d(\overline{a^4}, \overline{a^9})$ ,  $fi_d(\overline{a^5}, \overline{a^{10}})$ ,  $w_d^{(1,20)}(\overline{a^1}, \overline{a^2})$ .

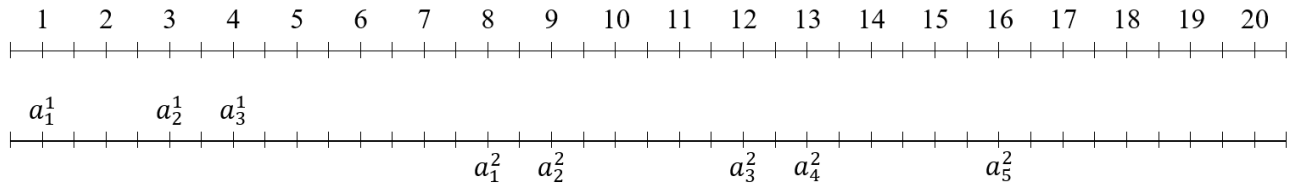


Рис. 2.1. Відношення *передую*

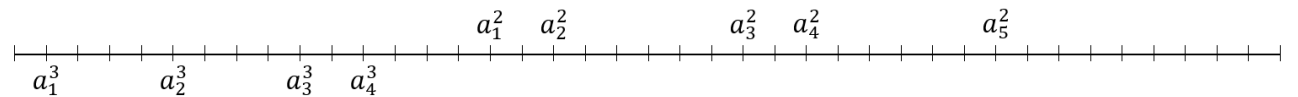


Рис. 2.2. Відношення *настає після*

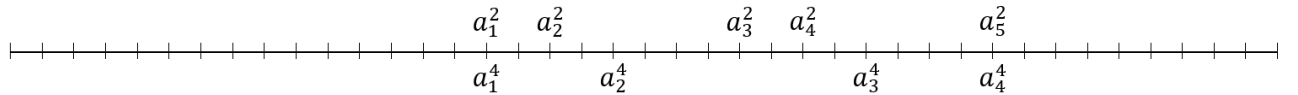


Рис. 2.3. Відношення *збігається з*

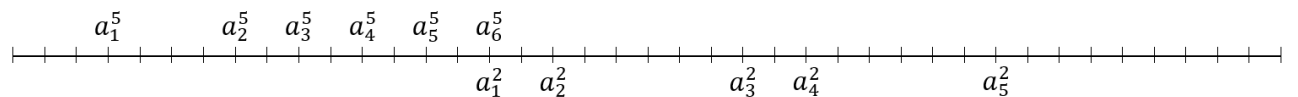


Рис. 2.4. Відношення *стикається з початком*

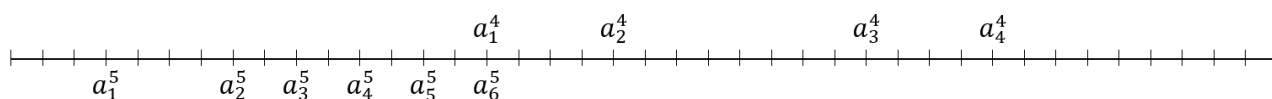


Рис. 2.5. Відношення *стикається з кінцем*

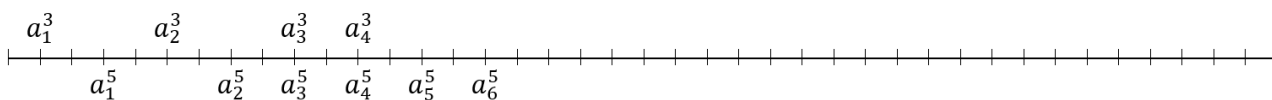


Рис. 2.6. Відношення *перекриває*

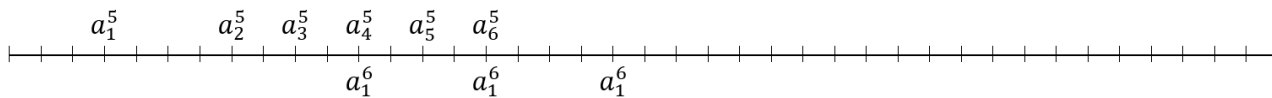


Рис. 2.7. Відношення *перекривається*

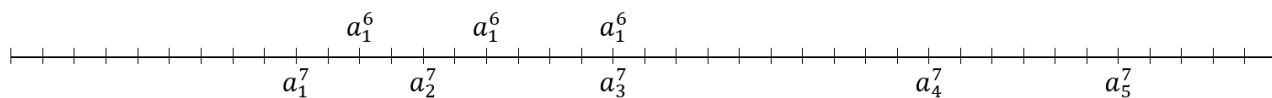


Рис. 2.8. Відношення *відбувається під час*

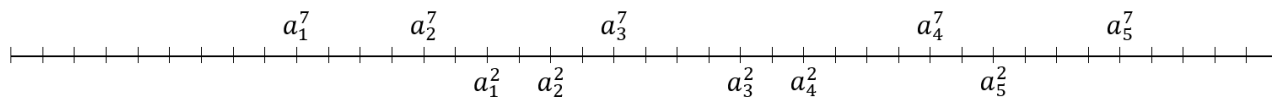


Рис. 2.9. Відношення *містить*

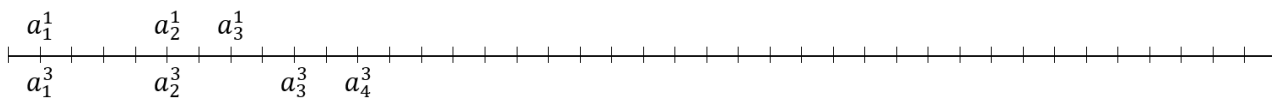


Рис. 2.10. Відношення *починає*

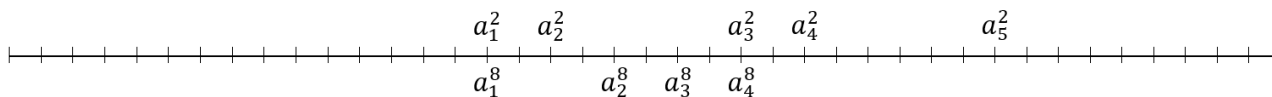


Рис. 2.11. Відношення *починається*

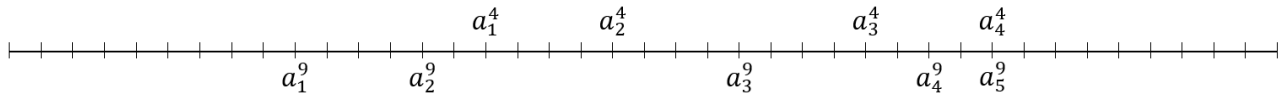


Рис. 2.12. Відношення *закінчує*

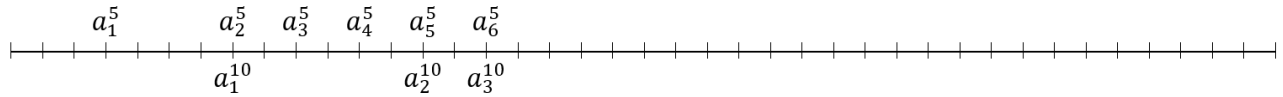


Рис. 2.13. Відношення *закінчується*

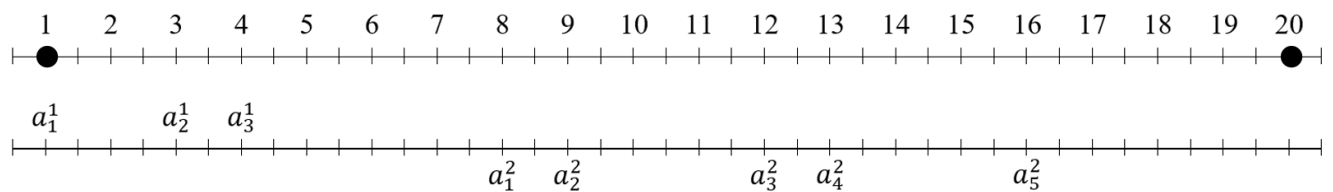


Рис. 2.14. Відношення *відбувається між*

Відношення між дискретними інтервалами призначені для виконання синхронізації темпоральних мультимодальних даних цифрових двійників.

### 2.3.3. Відношення між агрегатами

Відношення між агрегатами включають відношення між множинами агрегатів та відношення між кортежами агрегатів.

Відношеннями між множинами агрегатів є відношення: *еквівалентне* ( $\equiv$ ), *включає* ( $\supset$ ), *є включеним до* ( $\subset$ ). Результат цих відношень залежить від сумісності агрегатів.

Розглянемо агрегати  $A_1, A_2, A_3, A_4, A_5$ , and  $A_6$ , визначені (2.70).

$$\begin{aligned}
A_1 &= \llbracket M_1, M_2, \dots, M_N | \langle a_{i_1}^1 \rangle_{i_1=1}^{n_1^1}, \dots, \langle a_{i_N}^1 \rangle_{i_N=1}^{n_N^1} \rrbracket \\
A_2 &= \llbracket M_1, M_2, \dots, M_N | \langle a_{i_1}^2 \rangle_{i_1=1}^{n_1^2}, \dots, \langle a_{i_N}^2 \rangle_{i_N=1}^{n_N^2} \rrbracket \\
A_3 &= \llbracket M_1, M_2^3, \dots, M_S^3 | \langle a_{i_1}^3 \rangle_{i_1=1}^{n_1^3}, \dots, \langle a_{i_S}^3 \rangle_{i_S=1}^{n_S^3} \rrbracket \\
A_4 &= \llbracket M_1^4, \dots, M_W^4 | \langle a_{i_1}^4 \rangle_{i_1=1}^{n_1^4}, \dots, \langle a_{i_W}^4 \rangle_{i_W=1}^{n_W^4} \rrbracket \\
A_5 &= \llbracket M_1, M_2 | \langle a_{i_1}^5 \rangle_{i_1=1}^{n_1^5}, \langle a_{i_2}^5 \rangle_{i_2=1}^{n_2^5} \rrbracket \\
A_6 &= \llbracket M_2, M_1 | \langle a_{i_2}^6 \rangle_{i_2=1}^{n_2^6}, \langle a_{i_1}^6 \rangle_{i_1=1}^{n_1^6} \rrbracket
\end{aligned} \tag{2.70}$$

Отже, сумісність агрегатів (2.70) є наступною:

$$\begin{aligned}
A_1 \div A_2; \quad A_1 \div A_3; \quad A_1 \div A_5; \\
A_2 \div A_5; \quad A_1 \div A_4; \quad A_5 (\div) A_6.
\end{aligned}$$

Тоді можуть бути встановлені наступні відношення між цими агрегатами:

$$\{A_1\} \equiv \{A_2\}; \quad \{A_1\} \supset \{A_5\}; \quad \{A_5\} \subset \{A_2\}.$$

Решта відношень може бути визначена за допомогою заперечення:  $\{A_1\} \not\equiv \{A_3\}$ ,  $\{A_4\} \not\supset \{A_5\}$ ,  $\{A_3\} \not\subset \{A_6\}$ . Слід зазначити, що незважаючи на те, що  $\{A_1\} \not\supset \{A_3\}$ , ці агрегати мають спільну множину  $M_1$ . Для встановлення цього факту слід застосувати логічну операцію *перерізу*:  $\{A_1\} \cap \{A_3\} = M_1$  або у загальному випадку –  $\{A_p\} \cap \{A_q\} \neq \emptyset; p, q \in \mathbb{N}$ .

Визначимо відношення між множинами будь-яких двох агрегатів.

Відношення *еквівалентне* між двома агрегатами  $A_1$  та  $A_2$  визначається як:

$$\{A_1\} \equiv \{A_2\}, \text{ якщо } A_1 \div A_2 \tag{2.71}$$

Відношення *включає* між двома агрегатами  $A_1$  та  $A_2$  визначається як:

$$\{A_1\} \supset \{A_2\}, \text{ якщо } A_1 \doteq A_2, |A_1| > |A_2|, \{A_2\} = \langle M_1, \dots, M_K \rangle \text{ та } \langle M_1, \dots, M_K \rangle \in \{A_1\}. \quad (2.72)$$

Відношення *є включеним до* між двома агрегатами  $A_1$  та  $A_2$  визначається як:

$$\{A_1\} \subset \{A_2\}, \text{ якщо } A_1 \doteq A_2, |A_1| < |A_2|, \{A_1\} = \langle M_1, \dots, M_K \rangle \text{ та } \langle M_1, \dots, M_K \rangle \in \{A_2\}. \quad (2.73)$$

Відношення між кортежами агрегатів є ідентичними відношенням між окремими кортежами та включають відношення трьох категорій: арифметичні відношення, частотні відношення та інтервальні відношення. Проте, можливість їх застосування залежить від сумісності агрегатів: відношення між кортежами агрегатів можуть бути встановлені лише у випадку, коли ці агрегати є сумісними або квазісумісними. Приховано сумісні агрегати спочатку мають бути трансформовані у сумісні шляхом застосування до них операції *розміщення*, відповідно до Теорема 2.1. Крім того, інтервальні відношення можуть бути застосовані до дискретних інтервалів. Якщо кортежі, що мають перевірятись на будь-яке інтервальне відношення, не є дискретними інтервалами, вони мають бути перетворені на дискретні інтервали шляхом застосування операції *сортування*, відповідно до Теорема 2.2.

В разі необхідності встановлення відношень між окремими кортежами агрегатів це зазначається явно: наприклад,  $A_1(\overline{a^2}) \supset A_2(\overline{a^2})$ , якщо певне відношення правдиве для одного з кортежів кожного агрегата, або  $s_d(A_1(\overline{a^1}, \overline{a^3}), A_2(\overline{a^1}, \overline{a^3}))$  та  $A_1(\overline{a^1}, \overline{a^2}, \overline{a^5}) \ll A_2(\overline{a^1}, \overline{a^2}, \overline{a^5})$  якщо відношення правдиве для двох чи більше кортежів одночасно.



Розглянемо декілька прикладів. Нехай дані сумісні агрегати  $A_1$  та  $A_2$ :

$$A_1 = \llbracket M_1, M_2, M_3 \mid \langle 3, 4, 8, 9 \rangle, \langle 3, 1, 16, 12 \rangle, \langle 48, 13 \rangle \rrbracket;$$

$$A_2 = \llbracket M_1, M_2, M_3 \mid \langle 1, 5, 7, 8 \rangle, \langle 8, 10, 11, 12 \rangle, \langle 12, 15 \rangle \rrbracket;$$

Тоді між цими агрегатами та їх компонентами можна встановити такі відношення:  $\{A_1\} \equiv \{A_2\}$ ;  $\langle A_1 \rangle \sim \langle A_2 \rangle$ ;  $A_1(\overline{a^1}) \gg A_2(\overline{a^1})$ .

Нехай також дані квазісумісні агрегати  $A_3, A_4$ :

$$A_3 = \llbracket M_1, M_2 \mid \langle 8, 10, 11, 12 \rangle, \langle 17, 31 \rangle \rrbracket;$$

$$A_4 = \llbracket M_1, M_2, M_4 \mid \langle 2, 4, 8 \rangle, \langle 5, 7, 2, 6, 1 \rangle, \langle 1, 5, 7, 8 \rangle \rrbracket.$$

Ці агрегати пов'язані такими відношеннями:

$$\{A_3\} \subset \{A_4\}; \quad mi_d(A_3(\overline{a^1}), A_4(\overline{a^1})).$$

Всі значення у розглянутих прикладах є цілими числами. Проте, операції та відношення у АСА можуть також застосовуватись над будь-якими іншими типами даних, включаючи нечіткі значення.

Застосування нечітких значень має надзвичайну важливість для обробки темпоральних мультимодальних даних цифрових двійників, оскільки у значній частині практичних задач спостерігається невизначеність (нечіткість, неточність) при реєстрації окремих елементів даних та часу їх отримання.

Визначимо принципи виконання операцій та встановлення відношень АСА для нечітких значень.

## 2.4. Нечіткі значення в алгебраїчній системі агрегатів

Нечітке значення  $\tilde{a}_i$  є опуклою та нормалізованою нечіткою підмножиною множини  $\mathbb{R}$ , що характеризується функцією належності  $a_i(x)$ . Вибір функції належності [275–283] залежить від задач, що вирішуються.

Оскільки цю роботу присвячено вирішенню задачі обробки значень, що були виміряні або отримані шляхом оцінювання стану досліджуваного об'єкта, будемо використовувати функцію належності трикутного нечіткого числа, представлену у вигляді:

$$a_i(x) = \begin{cases} f_L & (a_i - \varepsilon_i) \leq x < a_i \\ 1 & x = a_i \\ f_R & a_i < x \leq (a_i + \varepsilon_i) \\ 0 & \text{в інших випадках,} \end{cases} \quad (2.74)$$

де  $\varepsilon_i$  – чітке значення похибки вимірювання;  $f_L = \frac{x - a_i + \varepsilon_i}{\varepsilon_i}$  та  $f_R = \frac{a_i + \varepsilon_i - x}{\varepsilon_i}$  – функції, що визначають лівий та правий край нечіткого числа відповідно.

Отже, визначимо трикутне нечітке число  $\tilde{a}_i$  трійкою значень  $((a_i - \varepsilon_i), a_i, (a_i + \varepsilon_i))$ . Тоді визначення агрегата (2.1) для нечіткого випадку має вигляд:

$$A = \left[ \left[ \tilde{M}_j | \langle (a_i^j - \varepsilon_i^j), a_i^j, (a_i^j + \varepsilon_i^j) \rangle_{i=1}^{n_j} \right]_{j=1}^N \right] \quad (2.75)$$

Очевидно, що (2.1) може бути отримано з (2.75) за умови, що  $\forall a_i^j(x) = 1$ . Крім того, можна припустити в (2.75), що  $\exists a_i^j(x) = 1$ , з чого випливає можливість комбінування у кортежі чітких та нечітких значень. Цей випадок також має практичне пояснення з точки зору задачі оброблення мультимодальних даних, оскільки він відповідає випадку, коли деякі значення були виміряні точно, а деякі значення були виміряні або оцінені з певною помилкою (наприклад, через людський фактор або технічні причини). Розширимо операції АСА на випадок, коли агрегат визначається (2.75).

Розглянемо принципи виконання операцій впорядкування агрегатів нечітких значень.

Серед всіх операцій впорядкування тип даних (чіткі або нечіткі) не впливає на результат виконання лише операції *розміщення*, оскільки у цій операції немає операцій над окремими елементами, а лише над кортежами елементів.

Визначимо нечіткий випадок для інших операцій впорядкування.

Нехай агрегат  $A$  має вигляд:

$$A = \llbracket \tilde{M}_1, \dots, \tilde{M}_k, \dots, \tilde{M}_N | \langle \tilde{a}_1^1, \dots, \tilde{a}_i^1, \dots, \tilde{a}_n^1 \rangle, \dots, \langle \tilde{a}_1^k, \dots, \tilde{a}_i^k, \dots, \tilde{a}_n^k \rangle, \dots, \langle \tilde{a}_1^N, \dots, \tilde{a}_i^N, \dots, \tilde{a}_n^N \rangle \rrbracket \quad (2.76)$$

$$\text{де } \tilde{a}_i^j = (a_i^j - \varepsilon_i^j, a_i^j, a_i^j + \varepsilon_i^j).$$

Застосуємо до агрегата (2.76) операції *сортування*, взявши за первісний кортеж  $k$ -тий кортеж цього агрегата. Тоді результатом операції *сортування за зростанням* над агрегатом  $A$  є агрегат  $S$ , такий що:

$$S = A \uparrow \langle \tilde{a}_1^k, \tilde{a}_2^k, \dots, \tilde{a}_i^k, \dots, \tilde{a}_n^k \rangle = \llbracket \tilde{M}_1, \dots, \tilde{M}_k, \dots, \tilde{M}_N | \langle \tilde{a}_\alpha^1, \tilde{a}_\beta^1, \dots, \tilde{a}_\nu^1, \dots, \tilde{a}_\omega^1 \rangle, \dots, \langle \tilde{a}_\alpha^k, \tilde{a}_\beta^k, \dots, \tilde{a}_\nu^k, \dots, \tilde{a}_\omega^k \rangle, \dots, \langle \tilde{a}_\alpha^N, \tilde{a}_\beta^N, \dots, \tilde{a}_\nu^N, \dots, \tilde{a}_\omega^N \rangle \rrbracket, \quad (2.77)$$

$$\text{де } \tilde{a}_\alpha^k \leq \tilde{a}_\beta^k \leq \dots \leq \tilde{a}_\nu^k \leq \dots \leq \tilde{a}_\omega^k, \quad a_m^j \in \langle \tilde{a}_i^j \rangle_{i=1}^{n_j}, \quad m \in [\alpha, \beta, \dots, \nu, \dots, \omega], \\ j = 1 \dots N.$$

Результатом операції *сортування за спаданням* над агрегатом  $A$  є агрегат  $S$ , такий що:

$$S = A \downarrow \langle \tilde{a}_1^k, \tilde{a}_2^k, \dots, \tilde{a}_i^k, \dots, \tilde{a}_n^k \rangle = \llbracket \tilde{M}_1, \dots, \tilde{M}_k, \dots, \tilde{M}_N | \langle \tilde{a}_\alpha^1, \tilde{a}_\beta^1, \dots, \tilde{a}_\nu^1, \dots, \tilde{a}_\omega^1 \rangle, \dots, \langle \tilde{a}_\alpha^k, \tilde{a}_\beta^k, \dots, \tilde{a}_\nu^k, \dots, \tilde{a}_\omega^k \rangle, \dots, \langle \tilde{a}_\alpha^N, \tilde{a}_\beta^N, \dots, \tilde{a}_\nu^N, \dots, \tilde{a}_\omega^N \rangle \rrbracket, \quad (2.78)$$

де  $\tilde{a}_\alpha^k \geq \tilde{a}_\beta^k \geq \dots \geq \tilde{a}_\nu^k \geq \dots \geq \tilde{a}_\omega^k$ ,  $a_m^j \in \langle \tilde{a}_i^j \rangle_{i=1}^{n_j}$ ,  $m \in [\alpha, \beta, \dots, \nu, \dots, \omega]$ ,  
 $j = 1 \dots N$ .

Отже, ключовою задачею при виконанні операцій впорядкування, зокрема, операцій *сортування*, є порівняння нечітких значень. Існує багато підходів до вирішення цієї задачі, включаючи дослідження [277–283]. Зокрема, в роботі [280], автори пропонують послідовний метод порівняння нечітких чисел, який ґрунтується на нечіткій арифметиці. В роботі [281], автори пропонують три методи нечіткого впорядкування (порівняння) значень: метод, що ґрунтується на застосуванні функції ранжування, метод, що ґрунтується на використанні опорної множини, та метод на основі попарного порівняння. В роботі [282], автори пропонують метод Fuzzy Max Order (FMO), що задає відношення часткового порядку на множині нечітких значень. Проте, згідно з роботою [283], FMO має обмеженість при порівнянні нечітких чисел, яку автори [283] пропонують виправити за допомогою методу Possibilistic Variation Order (PVO). Метод PVO розширює метод FMO шляхом ймовірнісної інтерпретації процедури впорядкування.

Будь-який з цих або інших підходів може бути застосований для порівняння нечітких значень під час виконання операцій впорядкування агрегатів виду (2.75).

Розглянемо агрегат  $A$  виду:

$$A = \llbracket \tilde{M}_1, \dots, \tilde{M}_k, \dots, \tilde{M}_N | \langle \tilde{a}_1^1, \dots, \tilde{a}_m^1, \dots, \tilde{a}_{m+p}^1, \tilde{a}_{m+p+1}^1, \dots, \tilde{a}_{n_1}^1 \rangle, \dots, \langle \tilde{a}_1^k, \dots, \tilde{a}_m^k, \dots, \tilde{a}_{m+p}^k, \tilde{a}_{m+p+1}^k, \dots, \tilde{a}_{n_k}^k \rangle, \dots, \langle \tilde{a}_1^N, \dots, \tilde{a}_m^N, \dots, \tilde{a}_{m+p}^N, \tilde{a}_{m+p+1}^N, \dots, \tilde{a}_{n_N}^N \rangle \rrbracket \quad (2.79)$$

де  $\tilde{a}_m^k = \tilde{a}_{m+1}^k = \dots = \tilde{a}_{m+p}^k$ ,

Тоді результат операції *проріджування* агрегата  $A$  за  $k$ -тим кортежем є агрегат  $S$ , такий що:

$$S = A \parallel \langle \tilde{a}_1^1, \dots, \tilde{a}_m^1, \dots, \tilde{a}_{m+p}^1, \tilde{a}_{m+p+1}^1, \dots, \tilde{a}_{n_1}^1 \rangle = \llbracket \tilde{M}_1, \dots, \tilde{M}_k, \dots, \tilde{M}_N | \langle \tilde{a}_1^1, \dots, \tilde{s}_m^1, \tilde{a}_{m+p+1}^1, \dots, \tilde{a}_{n_1}^1 \rangle, \dots, \langle \tilde{a}_1^k, \dots, \tilde{a}_m^k, \tilde{a}_{m+p+1}^k, \dots, \tilde{a}_{n_k}^k \rangle, \dots, \langle \tilde{a}_1^N, \dots, \tilde{a}_m^N, \tilde{a}_{m+p+1}^N, \dots, \tilde{a}_{n_N}^N \rangle \rrbracket \quad (2.80)$$

де  $\tilde{s}_m^1$  – це нечітке середнє.

Визначення 2.11. *Нечітким середнім* нечітких трикутних чисел будемо називати нечітке трикутне число  $\tilde{s} = (a - \hat{\varepsilon}, a, a + \hat{\varepsilon})$ , похибка  $\hat{\varepsilon}$  якого є середнім арифметичним похибок всіх нечітких трикутних чисел, для якого воно обчислюється:

$$\hat{\varepsilon} = average(\varepsilon_1, \dots, \varepsilon_n) = \frac{\sum_{i=1}^n \varepsilon_i}{n} \quad (2.81)$$

Таким чином, нечітке середнє  $\tilde{s}_m^1$  в (2.80) визначається як  $\tilde{s}_m^1 = (a_m^1 - \hat{\varepsilon}_m^1, a_m^1, a_m^1 + \hat{\varepsilon}_m^1)$ , де  $\hat{\varepsilon}_m^1 = average(\varepsilon_m^1, \varepsilon_{m+1}^1, \dots, \varepsilon_{m+p}^1) = \frac{\sum_{i=0}^p \varepsilon_{m+i}^1}{p+1}$ .

Операція *видалення*, яка дозволяє безумовно вилучити певний елемент з агрегата за його порядковим номером у кортежі, не потребує порівняння елементів кортежу, а отже, її результат не залежить від виду значень (чіткі або нечіткі), на відміну від операції *умовного видалення*, яка передбачає видалення елемента лише в разі виконання певної умови, що визначена через порівняння елементів, наприклад,  $S = A \bowtie \tilde{a}_m^k |_{\tilde{a}_m^k > \tilde{d}}$ , де  $\tilde{d}$  може бути як нечітким, так і чітким значенням.

Аналогічно, операція *вставлення*, яка дозволяє безумовно додати новий елемент до агрегата до або після певного елемента за його порядковим номером у кортежі, не потребує порівняння елементів кортежу та її результат не залежить від виду значень (чіткі або нечіткі). Водночас, операція *умовного вставлення* передбачає додавання елемента лише в разі виконання певної умови, що визначена через порівняння елементів, наприклад,  $S = A \rtimes (\tilde{d} < \tilde{a}_m^k) \big|_{\tilde{d} < \tilde{a}_m^k}$ , де  $\tilde{d}$  може бути як нечітким, так і чітким значенням.

Розглянемо принципи виконання логічних операцій над агрегатами нечітких значень.

На відміну від операції об'єднання над нечіткими множинами [278–279], яка ґрунтується на операції *тах*, операція *об'єднання* над кортежами нечітких значень в АСА враховує порядок слідування елементів у кортежах.

Нехай маємо кортеж  $\langle \tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_i, \dots, \tilde{a}_n \rangle$  та кортеж  $\langle \tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_i, \dots, \tilde{b}_n \rangle$ , тоді результатом операції *об'єднання* цих кортежів є кортеж виду:

$$\langle \tilde{a}_i \rangle_{i=1}^n \cup \langle \tilde{b}_i \rangle_{i=1}^n = \langle \tilde{a}_1, \tilde{a}_2, \dots, \tilde{a}_i, \dots, \tilde{a}_n, \tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_i, \dots, \tilde{b}_n \rangle, \quad (2.82)$$

в якому збережено порядок слідування елементів кортежів  $\langle \tilde{a}_i \rangle_{i=1}^n$  та  $\langle \tilde{b}_i \rangle_{i=1}^n$ .

Покажемо зв'язок операції *об'єднання* кортежів (2.82) та операція об'єднання нечітких множин. Для цього представимо кортежі  $\langle \tilde{a}_i \rangle_{i=1}^n$  та  $\langle \tilde{b}_i \rangle_{i=1}^n$  у вигляді:

$$\langle \tilde{a}_i \rangle_{i=1}^n = \langle (a_1, a_1(x)), \dots, (a_n, a_n(x)), (b_1, 0), \dots, (b_n, 0) \rangle, \quad (2.83)$$

$$\langle \tilde{b}_i \rangle_{i=1}^n = \langle (a_1, 0), \dots, (a_n, 0), (b_1, \mu(a_1)), \dots, (b_n, \mu(a_n)) \rangle$$

де  $a_i(x) \neq 0$ ,  $b_i(x) \neq 0$  – це функції належності нечітких чисел  $\tilde{a}_i$  та  $\tilde{b}_i$  відповідно. Тепер застосуємо функцію  $\max$ , на якій ґрунтується операція об'єднання нечітких множин, до відповідних елементів цих кортежів, наприклад,  $\max((a_1, a_1(x)), (a_1, 0)) = (a_1, a_1(x))$ , та отримаємо результат, що збігається з результатом операції об'єднання кортежів нечітких значень (2.82).

Застосуємо операцію об'єднання до квазісумісних агрегатів  $A$  та  $B$ , визначених як:

$$A = [\![\tilde{M}_1, \tilde{M}_1^1, \dots, \tilde{M}_k, \dots, \tilde{M}_N^1 | \langle \tilde{a}_1^1, \tilde{a}_2^1, \dots, \tilde{a}_n^1 \rangle, \langle \tilde{a}_1^2, \tilde{a}_2^2, \dots, \tilde{a}_n^2 \rangle, \dots, \langle \tilde{a}_1^k, \tilde{a}_2^k, \dots, \tilde{a}_n^k \rangle, \dots, \langle \tilde{a}_1^N, \tilde{a}_2^N, \dots, \tilde{a}_n^N \rangle]\!] \quad (2.84)$$

$$B = [\![\tilde{M}_1, \tilde{M}_2^2, \dots, \tilde{M}_k, \dots, \tilde{M}_N^2 | \langle \tilde{b}_1^1, \tilde{b}_2^1, \dots, \tilde{b}_n^1 \rangle, \langle \tilde{b}_1^2, \tilde{b}_2^2, \dots, \tilde{b}_n^2 \rangle, \dots, \langle \tilde{b}_1^k, \tilde{b}_2^k, \dots, \tilde{b}_n^k \rangle, \dots, \langle \tilde{b}_1^N, \tilde{b}_2^N, \dots, \tilde{b}_n^N \rangle]\!] \quad (2.85)$$

Тоді результатом об'єднання цих агрегатів є агрегат  $G$ , отриманий за таким правилом: якщо  $\langle \tilde{a}_i^k \rangle_{i=1}^n \in \tilde{M}_j$  та  $\langle \tilde{b}_i^k \rangle_{i=1}^n \in \tilde{M}_j$ , то елементи  $k$ -го кортежу агрегата  $B$  додаються у кінець  $k$ -го кортежу агрегата  $A$ , інакше, кортежі агрегата  $B$  додаються у кінець кортежу кортежів агрегата  $A$ , відповідні множини агрегата  $B$  додаються у кінець кортежу множин агрегата  $A$ :

$$G = A \cup B = [\![\tilde{M}_1, \tilde{M}_2^1, \dots, \tilde{M}_k, \dots, \tilde{M}_N^1, \tilde{M}_2^2, \dots, \tilde{M}_N^2 | \langle \tilde{a}_1^1, \tilde{a}_2^1, \dots, \tilde{a}_n^1, \tilde{b}_1^1, \tilde{b}_2^1, \dots, \tilde{b}_n^1 \rangle, \langle \tilde{a}_1^2, \tilde{a}_2^2, \dots, \tilde{a}_n^2 \rangle, \dots, \langle \tilde{a}_1^k, \tilde{a}_2^k, \dots, \tilde{a}_n^k, \tilde{b}_1^k, \tilde{b}_2^k, \dots, \tilde{b}_n^k \rangle, \dots, \langle \tilde{a}_1^N, \tilde{a}_2^N, \dots, \tilde{a}_n^N \rangle, \langle \tilde{b}_1^2, \tilde{b}_2^2, \dots, \tilde{b}_n^2 \rangle, \dots, \langle \tilde{b}_1^N, \tilde{b}_2^N, \dots, \tilde{b}_n^N \rangle]\!] \quad (2.86)$$

Подібно до операції *об'єднання*, операція *перерізу* кортежів та агрегатів виконується з оглядом на порядок слідування компонентів агрегатів, що відрізняє її від операції *перерізу* нечітких множин. Так, результатом операції *перерізу* кортежів нечітких значень є кортеж, який включає лише значення, що містяться в одному кортежі, які дорівнюють значенням з другого кортежу. Якщо дані кортежі нечітких значень  $\langle \tilde{a}_i \rangle_{i=1}^n$  та  $\langle \tilde{b}_i \rangle_{i=1}^n$ , то результатом застосування до них операції *перерізу* є кортеж виду:

$$\langle \tilde{a}_i \rangle_{i=1}^n \cap \langle \tilde{b}_i \rangle_{i=1}^n = \langle \tilde{a}_\alpha, \tilde{a}_\beta, \dots, \tilde{a}_\omega, \tilde{b}_\alpha, \tilde{b}_\beta, \dots, \tilde{b}_\omega \rangle, \quad (2.87)$$

де  $\langle \tilde{a}_p \rangle_{p=\alpha}^\omega \subset \langle \tilde{a}_i \rangle_{i=1}^n$ ,  $\langle \tilde{b}_q \rangle_{q=\alpha}^\omega \subset \langle \tilde{b}_i \rangle_{i=1}^n$ ;  $\forall \tilde{a}_p, \exists \tilde{b}_q$  та  $\forall \tilde{b}_q, \exists \tilde{a}_p$ , такі що  $\tilde{a}_p = \tilde{b}_q$ ;  $p, q \in [\alpha, \beta, \dots, \omega] \subset [1, 2, \dots, n]$ . Рівність нечітких значень  $\tilde{a}_p$  та  $\tilde{b}_q$  встановлюється порівнянням, як було розглянуто вище.

Застосування операції *перерізу* до агрегатів аналогічне застосуванню операції *перерізу* до кортежів. Нехай дані два квазісумісні кортежі  $A$  та  $B$ , визначені (2.84) та (2.85) відповідно. Тоді результатом операції *перерізу* цих агрегатів є агрегат  $G$ , отриманий застосуванням операції (2.87) до кожної пари кортежів  $\langle \tilde{a}_i^j \rangle_{i=1}^n \in \tilde{M}_j$  та  $\langle \tilde{b}_i^j \rangle_{i=1}^n \in \tilde{M}_j$ :

$$G = A \cap B = \llbracket \tilde{M}_1, \dots, \tilde{M}_k, \dots \mid \langle \tilde{a}_{\alpha_1}^1, \tilde{a}_{\beta_1}^1, \dots, \tilde{a}_{\omega_1}^1, \tilde{b}_{\alpha_1}^1, \tilde{b}_{\beta_1}^1, \dots, \tilde{b}_{\omega_1}^1 \rangle, \dots, \langle \tilde{a}_{\alpha_k}^k, \tilde{a}_{\beta_k}^k, \dots, \tilde{a}_{\omega_k}^k, \tilde{b}_{\alpha_k}^k, \tilde{b}_{\beta_k}^k, \dots, \tilde{b}_{\omega_k}^k \rangle, \dots \rrbracket \quad (2.88)$$

де  $\langle \tilde{a}_p^j \rangle_{p=\alpha_j}^{\omega_j} \subset \langle \tilde{a}_i^j \rangle_{i=1}^{n_j}$ ,  $\langle \tilde{b}_q^j \rangle_{q=\alpha_j}^{\omega_j} \subset \langle \tilde{b}_i^j \rangle_{i=1}^{n_j}$ ;  $\forall \tilde{a}_p^j, \exists \tilde{b}_q^j$  та  $\forall \tilde{b}_q^j, \exists \tilde{a}_p^j$ , такі що  $\tilde{a}_p^j = \tilde{b}_q^j$ ;  $p, q \in [\alpha_j, \beta_j, \dots, \omega_j] \subset [1, 2, \dots, n_j]$ .



Нечіткий випадок інших логічних операцій АСА, які є різниця, симетрична різниця та виключний переріз, можна визначити на основі аналогічного підходу.

## 2.5. Порівняльний аналіз алгебраїчної системи агрегатів

Порівняємо можливості з опису та обробки темпоральних мультимодальних даних, які надає АСА та інші математичні концепції, які можуть бути застосовані для вирішення задач, що передбачають такі дані.

Аналізом даних, які асоційовані з поняттям часу, а отже, й порядку слідування даних, займається теорія часових рядів (ТЧР) [284–286]. *Часовим рядом* називають послідовність значень  $x_t$ , кожне з яких отримано у певний момент часу  $t$ . Теорія часових рядів надає математичний апарат для аналізу послідовностей даних, дослідження трендів та передбачення, проте, часові ряди не передбачають застосування до них логічних та інших операцій, які б дозволяли логічну обробку темпоральних мультимодальних даних цифрових двійників.

Теорія множин (ТМ) [262–264, 287] та теорія мультимножин (ТММ) [288–294] надають гнучкі механізми логічної обробки сукупностей даних, проте вони не передбачають врахування порядку слідування елементів та, відповідно, не надають можливості впорядкування елементів сукупності темпоральних мультимодальних даних цифрового двійника.

Такою абстракцією, як кортеж, оперує алгебра кортежів [295, 296]. Алгебра кортежів (АК) – це алгебраїчна система, носієм якої є довільна сукупність багатомісних відношень, виражених у специфічних структурах (елементарний кортеж, С-кортеж, С-система, D-кортеж, D-система), що називаються АК-об'єктами [295, 296]. Основні операції АК включають операції алгебри множин та операції з атрибутами. В АК використовуються

три операції алгебри множин: об'єднання, переріз та доповнення. Операціями з атрибутами є перейменування, перестановка, елімінація та додавання фіктивного атрибута. Для порівняння АК-об'єктів використовуються два основних відношення: включення та рівність. Аналіз апарату АК дозволяє зробити висновок, що хоча АК дозволяє агрегацію даних, вона не надає достатньо широкого спектру операцій та відношень для вирішення задач формальної специфікації та ефективної обробки темпоральних мультимодальних даних цифрових двійників.

Порівняльний аналіз АСА та розглянутих математичних концепцій показав, що лише АСА у повній мірі задовольняє вимогам до теоретичного апарату для формального опису цифрових двійників досліджуваних об'єктів. Серед інших математичних концепцій, які частково задовольняють висунутим вимогам, для опису цифрових двійників також може бути застосований апарат АК. Результати порівняння властивостей АСА з розглянутими математичними концепціями наведено у табл. 2.2.

Таблиця 2.2 – Порівняльний аналіз АСА

Властивості	ТЧР	ТМ	ТММ	АК	АСА
Операції впорядкування	так	ні	ні	так	так
Логічні операції	ні	так	так	так	так
Нечіткі значення	частково	частково	частково	частково	так
Агрегація даних	ні	ні	частково	так	так
Темпоральні дані	так	ні	ні	ні	так

Таким чином, алгебраїчна система агрегатів надає необхідний апарат для формальної специфікації цифрового двійника досліджуваного об'єкта. Також цей апарат може бути використаний при розробленні алгоритмічно-програмного забезпечення для обробки темпоральних мультимодальних даних цифрового двійника.

## **2.6. Висновки до розділу 2**

У цьому розділі вирішувалась наукова задача розроблення теоретичних основ для формального опису цифрового двійника досліджуваного об'єкта. В результаті дослідження отримано такі результати.

1. Сформульовано вимоги до теоретичного апарату для формальної специфікації цифрового двійника досліджуваного об'єкта. Виявлено, що існуючі математичні концепції не задовольняють у повній мірі цим вимогам, оскільки не надають теоретичний апарат для опису темпоральних мультимодальних даних.

2. Розроблено апарат алгебраїчної системи агрегатів, носієм якої є непуста множина математичних об'єктів – агрегатів, які являють собою впорядковані скінчені сукупності елементів. Особливістю цієї алгебраїчної системи є врахування порядку слідування елементів (кортежів, агрегатів) при виконанні операцій, введенням операцій впорядкування та введенням спеціальних відношень (частотних, інтервальних) між елементами. Властивості алгебраїчної системи агрегатів дозволяють використовувати її апарат для формального опису цифрових двійників досліджуваних об'єктів і для розроблення методів обробки темпоральних мультимодальних даних.

3. Розроблено математичний апарат для обробки дискретних інтервалів, який є розвитком інтервальної алгебри Аллена. На відміну від інтервалів Аллена дискретні інтервали та дискретні інтервали без

повторень дозволяють визначати дискретні події, що дозволяє використовувати визначені в алгебраїчній системі агрегатів інтервальні відношення при розроблені алгоритмічно-програмного забезпечення для синхронізації темпоральних мультимодальних даних.

4. Застосовано принципи нечіткої логіки до операцій та відношень алгебраїчної системи агрегатів, що дозволяє розроблювати алгоритмічно-програмне забезпечення для синхронізації нечітко визначених темпоральних мультимодальних даних.

### РОЗДІЛ 3. МОДЕЛІ ПОДАННЯ ТА МЕТОДИ ОБРОБКИ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВОГО ДВІЙНИКА ДОСЛІДЖУВАНОВОГО ОБ'ЄКТА

Розроблені у попередньому розділі теоретичні основи подання темпоральних мультимодальних даних мають на меті створення моделей подання досліджуваного об'єкта у цифровому вигляді. Розглянемо застосування апарату алгебраїчної системи агрегатів для формальної специфікації цифрового двійника досліджуваного об'єкта.

#### 3.1. Концепція мультиобразу для формальної специфікації моделей цифрового двійника

Розглянемо об'єкт спостереження  $S$ , який виявляє свою сутність через набір властивостей  $F_1, F_2, \dots, F_N$ , які можуть бути виміряні. В багатьох випадках ці властивості є взаємопов'язаними, оскільки вони представляють стан одного і того ж об'єкта та, фактично, є різними проявами його поведінки, що змінюється з плином часу. Наприклад, згідно з дослідженнями фізіології людини [297], при підвищенні температури тіла пацієнта, його пульс також зростає. Комплексне подання даних, що відображають характеристики досліджуваного об'єкта, сприяє кращому розумінню загальної картини поведінки цього об'єкта.

Представимо виміряні значення властивостей досліджуваного об'єкта  $S$  як агрегат  $A_S$ , компонентами якого є кортежі значень  $\langle f_i^j \rangle_{i=1}^{n_j}$  властивостей  $F_j$  ( $j = 1 \dots N$ ) об'єкта  $S$ :

$$A_S = \llbracket M_1, M_2, \dots, M_N | \langle f_1^1, f_2^1, \dots, f_{n_1}^1 \rangle, \langle f_1^2, f_2^2, \dots, f_{n_2}^2 \rangle, \dots, \langle f_1^N, f_2^N, \dots, f_{n_N}^N \rangle \rrbracket.$$

Якщо вимірювання значень різних властивостей є одночасним, то всі кортежі агрегата  $A_S$  мають однакову довжину. В цьому випадку, агрегат  $A_S$  може бути розглянутий як послідовність мультикомпонентних значень (рис. 3.1).

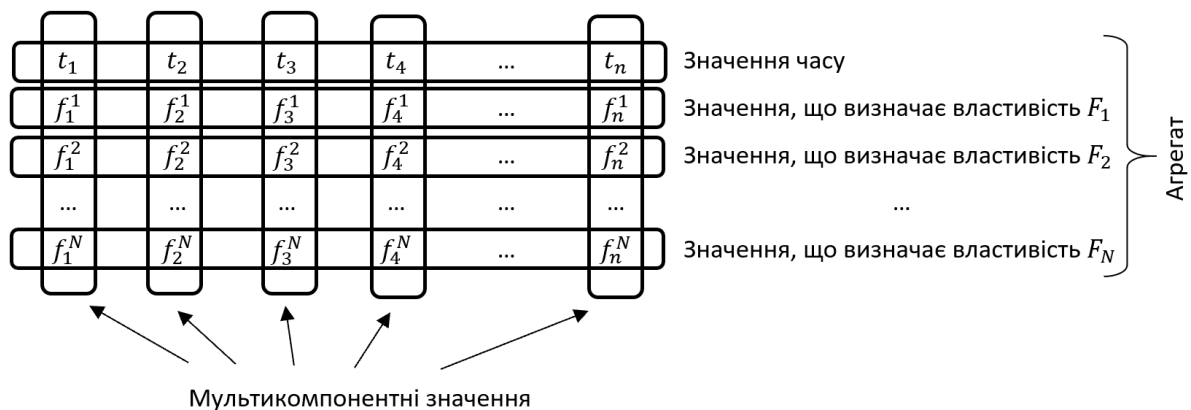


Рис. 3.1. Способи подання агрегованих даних

Ця послідовність мультикомпонентних значень може бути розглянута як функція багатьох змінних:  $y = x(t, F_1, F_2, \dots, F_N)$ . Такий погляд на агреговані дані, дозволяє застосовувати до них апарат числення багатьох змінних у задачах, в яких є доречними подання та обробка даних за допомогою операцій та відношень, визначених в АСА, тобто застосування апарату АСА не заперечує використання інших математичних концепцій.

Якщо дані про об'єкт спостереження отримані з урахуванням часу спостереження або вимірювання, то у агрегаті має бути присутнім кортеж відповідних темпоральних значень, які відповідають моментам часу, коли значення кортежів мультимодальних даних були отримані. Будемо називати такий агрегат *мультиобразом* цифрового двійника досліджуваного об'єкта.

Мультиобраз надає формальний опис послідовностей мультимодальних даних про досліджуваний об'єкт, отриманих з

урахуванням часу в результаті вимірювання, генерації та інших операцій з отримання даних.

Фактично, мультиобраз є абстрактною специфікацією цифрового двійника, на основі якої можуть бути побудовані різноманітні моделі цифрового двійника. Визначимо поняття мультиобразу формально.

Визначення 3.1. *Мультиобраз* – це непустий агрегат виду:

$$I = \llbracket T, M_1, \dots, M_N | \langle t_1, \dots, t_\tau \rangle, \langle a_1^1, \dots, a_{n_1}^1 \rangle, \dots, \langle a_1^N, \dots, a_{n_N}^N \rangle \rrbracket, \quad (3.1)$$

де  $T$  – це множина значень часу;  $\tau \geq n_j, j \in [1, \dots, N]$ .

Отже, у математичному сенсі, *мультиобразом* є агрегат, першим кортежем якого є непустий кортеж значень часу. Ці значення можуть бути натуральними числами або будь-якими іншими значеннями, які забезпечують очевидність та однозначність подання інформації про моменти часу, в які було отримано елементи інших кортежів мультиобразу.

Оскільки мультиобраз є агрегатом, то до нього можуть бути застосовані всі операції та відношення, визначені в АСА. При цьому потрібно брати до уваги, що мультиобрази, на відміну від агрегатів, завжди є квазісумісними або сумісними. Доведемо це.

Теорема 3.1 (теорема про достатню умову квазісумісності агрегатів).

Якщо агрегати  $I_1$  та  $I_2$  є мультиобразами, то  $I_1 \doteq I_2$ .

Доведення:

Нехай дані мультиобрази

$$I_1 = \llbracket T, M_1^1, \dots, M_N^1 | \langle t_i^a \rangle_{i=1}^{\tau_1}, \langle a_i^1 \rangle_{i=1}^{n_1}, \dots, \langle a_i^N \rangle_{i=1}^{n_N} \rrbracket \text{ та}$$

$$I_2 = \llbracket T, M_1^2, \dots, M_N^2 | \langle t_i^b \rangle_{i=1}^{\tau_2}, \langle b_i^1 \rangle_{i=1}^{n_1}, \dots, \langle b_i^N \rangle_{i=1}^{n_N} \rrbracket.$$

За визначенням мультиобразу (Визначення 3.1) мультиобразом є агрегат, першим кортежем якого є непустий кортеж значень часу. Отже, мультиобрази  $I_1$  та  $I_2$  одночасно є агрегатами. З'ясуємо їх сумісність. Для цього застосуємо операцію перерізу до кортежів множин агрегатів  $I_1$  та  $I_2$ :  $\{I_1\} \cap \{I_2\} = T \neq \{I_1\} \neq \{I_2\}$ . З цього випливає, що  $I_1$  та  $I_2$  є квазісумісними, оскільки згідно з Визначенням 2.6 (умови (2.7)) два агрегати є квазісумісними, якщо  $\{I_1\} \not\equiv \{I_2\}$  та  $\{I_1\} \cap \{I_2\} \neq \emptyset$ .

Оскільки для виконання умов (2.7) у кортежі множин агрегатів  $I_1$  та  $I_2$  має бути принаймні одна спільна множина, то *достатньою умовою* квазісумісності агрегатів  $I_1$  та  $I_2$  є те, що ці агрегати є мультиобразами.

Що і потрібно було довести. ■

Наслідок з теореми 3.1. Оскільки сумісність може розглядатись як окремий випадок квазісумісності, то з Теореми 3.1 випливає:  $\exists I_1$  та  $\exists I_2$ , такі що  $I_1 \div I_2$ .

Застосуємо концепцію мультиобразу для формальної специфікації мультимедійної візуальної моделі цифрового двійника, що визначається візуальними даними, аудіо-даними, тактильними даними, температурними даними та даними про оточуюче середовище, а також нюховими даними.

Монофонічний цифровий аудіо-сигнал може бути представлений як кортеж виду:

$$\bar{S}_{mono} = \langle s_k \rangle_{k=0}^K = \langle s_0, s_1, \dots, s_K \rangle, \quad (3.2)$$

де  $s_k \in R$  – значення деякої фізичної величини, яка визначає стан середовища під час розповсюдження акустичної хвилі, наприклад, звуковий тиск.



Стереофонічний цифровий аудіо-сигнал може бути визначений таким кортежем:

$$\bar{S}_{stereo} = \langle \langle s_k^1, s_k^2 \rangle \rangle_{k=0}^K = \langle \langle s_0^1, s_0^2 \rangle, \langle s_1^1, s_1^2 \rangle, \dots, \langle s_K^1, s_K^2 \rangle \rangle. \quad (3.3)$$

де  $s_k^1$  – значення рівня звукового тиску для лівого каналу;  $s_k^2$  – значення рівня звукового тиску для правого каналу;  $s_k^{1,2} \in R$ .

Багатоканальний цифровий аудіо-сигнал може бути визначений наступним кортежем:

$$\begin{aligned} \bar{S}_{multi} = \langle \langle s_k^1, s_k^2, \dots, s_k^Q \rangle \rangle_{k=0}^K = \\ \langle \langle s_0^1, s_0^2, \dots, s_0^Q \rangle, \dots, \langle s_K^1, s_K^2, \dots, s_K^Q \rangle \rangle \end{aligned} \quad (3.4)$$

де  $s_k^q$  – це значення рівня звукового тиску для  $q$ -го каналу;  $q \in [1, \dots, Q]$ ;  $s_k^q \in R$ .

Двовимірне зображення розміром  $N \times M$  пікселів може бути описане як кортеж виду:

$$\bar{C}_{2D} = \langle \langle c_n^1, c_n^2, \dots, c_n^M \rangle \rangle_{n=0}^N = \langle \langle c_0^1, \dots, c_0^M \rangle, \dots, \langle c_N^1, \dots, c_N^M \rangle \rangle. \quad (3.5)$$

де  $c_n^m$  – це кортеж кольірних компонент пікселя;  $c_n^m \in Z^3$ ;  $m \in [1, \dots, M]$ .

Кортеж (3.5) являє собою кортеж кортежів,  $n$ -й елемент якого – це набір значень яскравості кольору пікселів  $n$ -го рядка зображення. В свою чергу  $c_n^m$  може бути представлений як кортеж кольірних компонент у певній кольірній моделі. Наприклад, у моделі RGB [298]  $c_n^m$  визначається як кортеж виду:

$$c_n^m = \langle r_n^m, g_n^m, b_n^m \rangle \quad (3.6)$$

де  $r_n^m, g_n^m, b_n^m$  – колірні компоненти в моделі RGB;  $r_n^m, g_n^m, b_n^m \in Z$ .

Формальне визначення тривимірного зображення залежить від технології його відтворення [39, 42]. Наприклад, воксельна графіка [15] передбачає визначення зображення розміром  $N \times M \times K$  пікселів, яке може бути описане як кортеж виду:

$$\bar{C}_{3D} = \langle \langle \langle c_{n,k}^1 \rangle_{k=1}^K, \langle c_{n,k}^2 \rangle_{k=1}^K, \dots, \langle c_{n,k}^M \rangle_{k=1}^K \rangle \rangle_{n=1}^N \quad (3.7)$$

де  $c_{n,k}^m$  – це трійка колірних компонент,  $c_{n,k}^m \in Z^3$ ;  $m \in [1, \dots, M]$ .

Якщо тривимірне зображення створене за технологією стереовізуалізації [39, 42], то воно може бути описане як кортеж виду:

$$\bar{C}_{stereo} = \langle \langle \langle l_n^1, l_n^2, \dots, l_n^M \rangle, \langle r_n^1, r_n^2, \dots, r_n^M \rangle \rangle \rangle_{n=0}^N \quad (3.8)$$

де  $l_n^m$  – компонент даних лівого фрейму;  $r_n^m$  – компонент даних правого фрейму.

Двовимірне відео може бути подане як агрегат, що включає кортеж аудіо-даних та кортеж двовимірних зображень:

$$V_{2D} = \llbracket R^Q, Z^3 \mid \bar{S}, \bar{C} \rrbracket \quad (3.9)$$

де  $\bar{S}$  – це  $\bar{S}_{mono}$ ,  $\bar{S}_{stereo}$  або  $\bar{S}_{multi}$ ;  $\bar{C}$  – це  $\bar{C}_{2D}$  або  $\bar{C}_{3D}$ .

Тривимірне відео може бути подане як агрегат послідовності стереозображень та просторового аудіо ( $Q$ -канальний аудіо-сигнал):

$$V_{3D} = \llbracket R^Q, Z^3 \mid \bar{S}_{multi}, \bar{C}_{stereo} \rrbracket \quad (3.10)$$

Тактильні дані, отримані за допомогою рукавички віртуальної реальності [41, 42], можуть бути подані як кортеж значень інтенсивності руху кожного пальця. Тоді, якщо рух кожного пальця реєструється одним давачем, можна визначити кортеж тактильних даних наступним чином:

$$\bar{H} = \langle \langle h_k^1, h_k^2, h_k^3, h_k^4, h_k^5 \rangle \rangle_{k=0}^K \quad (3.11)$$

де  $h_k^i$  – це значення інтенсивності руху;  $h_k^i \in R$ ;  $i \in [1, \dots, 5]$ .

Температурні дані та інші дані про оточуюче середовище можуть бути формалізовані так:

$$\bar{E} = \langle \langle e_k^T, e_k^H, e_k^W \rangle \rangle_{k=0}^K \quad (3.12)$$

де  $e_k^T$  – це значення температури,  $e_k^T \in R$ ;  $e_k^H$  – значення вологості повітря,  $e_k^H \in R$ ;  $e_k^W$  – значення сили вітру,  $e_k^W \in R$ .

Нюхові дані можуть бути визначені як наступний кортеж кортежів:

$$\bar{O} = \langle \langle o_k^T, o_k^I, o_k^d, o_k^s \rangle \rangle_{k=0}^K \quad (3.13)$$

де  $o_k^T$  – тип запаху згідно з певною моделлю нюхових даних [145, 147],  $o_k^I$  – інтенсивність запаху [149],  $o_k^d \in R$  – тривалість дії нюхових подразників;  $o_k^s \in R$  – коефіцієнт асиметрії синхронізації нюхових даних [149].

Тоді, мультиобраз, що задає мультимедійну візуальну модель цифрового двійника фізичного об'єкта, може бути визначений наступним чином:

$$I = \llbracket T, R^Q, Z^3, R^5, R^3, \langle M_O, M_I, R^2 \rangle | \langle t_1, \dots, t_\tau \rangle, \bar{S}_{multi}, \bar{C}_{stereo}, \bar{H}, \bar{E}, \bar{O} \rrbracket \quad (3.14)$$

Сформулюємо метод створення мультиобразу досліджуваного об'єкта для формальної специфікації його цифрового двійника.

### 3.2. Метод створення мультиобразу досліджуваного об'єкта

Вхідними даними для метода створення мультиобразу є набори темпоральних даних та тип (модальність) даних кожного набору. Метод складається з сімох етапів. Результатом методу є мультиобраз досліджуваного об'єкта, поданий у вигляді впорядкованої сукупності темпоральних мультимодальних даних.

Першим етапом методу є формування структури даних мультиобразу досліджуваного об'єкта, яка виконується, виходячи із задачі, що вирішується. Мультиобраз може визначатись математично, згідно з (3.1), або будь-яким іншим способом, який дозволяє однозначно задати порядок слідування наборів даних та їх модальність.

Наприклад, альтернативою математичному визначенню мультиобразу (3.14) може слугувати схематичне визначення цього мультиобразу, показане на рис. 3.2, у кожному блоці якого наведено назву послідовності (кортежу) даних, модальність даних (множина, до якої вони належать) та довжина кортежу даних.

На другому етапі методу відбувається декомпозиція специфікації мультиобразу на набір специфікацій *часткових* мультиобразів виду:

$$I_j = \llbracket T, M_j | \langle t_1, \dots, t_{\tau_j} \rangle, \langle a_1, \dots, a_{n_j} \rangle \rrbracket \quad (3.15)$$

де  $T$  – це множина значень часу;  $M_j$  – множина даних  $j$ -ї модальності;  $\tau_j, n_j \in \mathbb{N}$ ;  $\tau_j \geq n_j$ ;  $j \in [1, \dots, N]$ ;  $N$  – кількість наборів даних.



Рис. 3.2. Схематична специфікація мультиобразу

На третьому етапі відбувається отримання даних. Процедура отримання даних передбачає визначення способу (протоколу, формату) передавання даних, визначення періоду часу для передавання даних, встановлення зв'язку з джерелом даних та, власне, отримання даних визначеним способом та у визначеному форматі.

На четвертому етапі відбувається підготовка даних кожної модальності з метою формування відповідного часткового мультиобразу. Частковий мультиобраз являє собою агрегат, який включає два кортежі елементів: кортеж часових значень та кортеж елементів певної модальності.

Елементами є окремі значення або впорядковані сукупності значень (однорідні або різнорідні). Прикладом кортежу, який складається з окремих значень, є кортеж значень рівня звукового тиску (3.2). Прикладами кортежу, який складається з однорідних кортежів, можуть слугувати кортеж, що визначає двовимірне зображення (3.5), кортеж тактильних даних (3.11) тощо. Прикладами кортежу, який складається з різнорідних кортежів, можуть слугувати кортеж даних про оточуюче середовище (3.12) та кортеж нюхових даних (3.13). Підготовка даних, яка виконується на цьому етапі, полягає у визначенні (виявленні) значень часу у сукупності даних, отриманих на попередньому етапі. Ця процедура може бути як тривіальною, коли формат даних передбачає явне подання значень часу для компонента темпоральних даних, так і складною, коли значення часу подані у неявній формі, або є нечіткими. Спосіб виявлення прихованих або нечітко визначених значень часу має розроблятися для кожного конкретного формату подання даних.

На п'ятому етапі відбувається об'єднання часткових мультиобразів у єдиний мультиобраз. Процедура об'єднання мультиобразів включає наступні кроки.

1. Нормалізація часткових мультиобразів (*нормалізованим* мультиобразом (*нормалізованим* кортежем) називатимемо мультиобраз (кортеж) до елементів якого додані фіктивні елементи; *фіктивним* елементом називатимемо деяке унікальне значення, яке відсутнє у кортежі до виконання нормалізації; як фіктивний може використовуватись пустий елемент  $\emptyset$ , невизначений елемент  $\_$  або будь-яке інше унікальне значення; на інших етапах роботи з мультиобразом (аналіз, відтворення тощо) фіктивні елементи ігноруються):

$$\begin{aligned}\hat{I}_j = I_j \bowtie \left( \langle \emptyset \rangle_{k=1}^{E_i^j} \succ a_{n_j}^j \right) &= \left[ \left[ T, M_j | \langle t_i \rangle_{i=1}^{\tau_j}, \langle \langle a_{i_j}^j \rangle_{i_j=1}^{n_j}, \langle \emptyset \rangle_{l=1}^{E_i^j} \rangle \right] \right] = \\ &= \left[ \left[ T, M_j | \langle t_i \rangle_{i=1}^{\tau_j}, \langle \hat{a}_{i_j}^j \rangle_{i_j=1}^{n_j+E_i^j} \right] \right]\end{aligned}\quad (3.16)$$

де  $\bowtie$  – операція вставлення;

$E_i^j$  – кількість фіктивних елементів;  $E_i^j = (\sum_{j=1}^N \tau_j) - n_j$ ;

$\hat{a}_{i_j}^j$  – елемент нормалізованого кортежу;  $j \in [1, \dots, N]$ .

2. Об'єднання нормалізованих часткових мультиобразів:

$$I = \cup_{j=1}^N \hat{I}_j = \left[ \left[ T, M_1, \dots, M_N | \langle \langle t_i \rangle_{i=1}^{\tau_j} \rangle_{j=1}^N, \langle \hat{a}_{i_1} \rangle_{i_1=1}^{n_1+E_i^1}, \dots, \langle \hat{a}_{i_N} \rangle_{i_N=1}^{n_N+E_i^N} \right] \right]. \quad (3.17)$$

На шостому етапі відбувається сортування мультиобразу, отриманого на попередньому етапі, за кортежем часових значень:

$$I_S = I \uparrow \bar{t} = \left[ \left[ T, M_1, \dots, M_N | \langle \langle t_\sigma \rangle_{\sigma=1}^{\tau_j} \rangle_{j=1}^N, \langle \hat{a}_{\sigma_1} \rangle_{\sigma_1=1}^{n_1+E_i^1}, \dots, \langle \hat{a}_{\sigma_N} \rangle_{\sigma_N=1}^{n_N+E_i^N} \right] \right], \quad (3.18)$$

де  $\sigma$  та  $\sigma_j$  – індекси, що задають порядок слідування елементів у відсортованому кортежі;  $j \in [1, \dots, N]$ .

На сьомому етапі відбувається проріджування відсортованого мультиобразу, отриманого на попередньому етапі, за кортежем часових значень:

$$\begin{aligned}I_S = I \parallel \bar{t} &= \left[ \left[ T, M_1, \dots, M_N | \langle t_\sigma \rangle_{\sigma=1}^{(\sum_{j=1}^N \tau_j) - \delta}, \langle \hat{a}_{\sigma_1} \rangle_{\sigma_1=1}^{n_1+E_i^1 - \delta}, \dots, \right. \right. \\ &\quad \left. \left. \langle \hat{a}_{\sigma_N} \rangle_{\sigma_N=1}^{n_N+E_i^N - \delta} \right] \right],\end{aligned}\quad (3.19)$$

де  $\delta$  – кількість відкинутих дублікатів значень часу.

Виконати аналіз запропонованого методу створення мультиобразу досліджуваного об'єкта можна лише якісно, оскільки інші методи створення мультиобразу наразі не існують.

Розглянемо приклад, який демонструє виконання основних етапів методу. Припустимо, що пацієнт, який перебуває вдома під дистанційним медичним наглядом, за призначенням лікаря має тричі на день вимірювати температуру тіла та частоту пульсу: вранці, вдень та ввечері. В результаті проведеного самообстеження пацієнт надав лікарю запитувані дані вимірювань, супроводжуючи їх нечіткими значеннями часу, які можна представити у вигляді двох часткових мультиобразів – часткового мультиобразу  $A$ , що містить значення температури, та часткового мультиобразу  $B$ , що містить значення пульсу:

$$A = \llbracket \tilde{T}, M_t \mid \langle (8:30, 9:00, 9:30), (13:30, 14:00, 14:30), (18:45, 19:00, 19:15) \rangle, \langle 36.6, 36.9, 37.1 \rangle \rrbracket$$

$$B = \llbracket \tilde{T}, M_p \mid \langle (8:45, 9:00, 9:15), (13:45, 14:00, 14:15), (18:50, 19:00, 19:10) \rangle, \langle 75, 78, 80 \rangle \rrbracket.$$

Для отримання мультиобразу, який відображає дані цього дослідження, потрібно об'єднати часткові мультиобрази  $A$  та  $B$ . Безпосереднє застосування операції об'єднання до цих часткових мультиобразів призвело би до того, що кортеж часових значень об'єднаного агрегата  $G$  був би вдвічі довший, ніж кортеж значень температури та кортеж значень пульсу, що зруйнувало би відповідність між значеннями часу та вимірюваними значеннями. Щоб уникнути цієї суперечності, спочатку потрібно виконати нормалізацію цих часткових мультиобразів, а саме, застосувати операцію вставлення і додати фіктивні елементи у кінець кортежу значень температури та у кінець кортежу значень пульсу:



$$A_1 = \llbracket \tilde{T}, M_t \mid \langle (8:30, 9:00, 9:30), (13:30, 14:00, 14:30), (18:45, 19:00, 19:15) \rangle, \langle 36.6, 36.9, 37.1, \emptyset, \emptyset, \emptyset \rangle \rrbracket$$

$$B_1 = \llbracket \tilde{T}, M_p \mid \langle (8:45, 9:00, 9:15), (13:45, 14:00, 14:15), (18:50, 19:00, 19:10) \rangle, \langle 75, 78, 80, \emptyset, \emptyset, \emptyset \rangle \rrbracket.$$

Після нормалізації застосуємо до цих агрегатів операцію об'єднання:

$$G_1 = A_1 \cup B_1 = \llbracket \tilde{T}, M_t, M_p \mid \langle (8:30, 9:00, 9:30), (13:30, 14:00, 14:30), (18:45, 19:00, 19:15), (8:45, 9:00, 9:15), (13:45, 14:00, 14:15), (18:50, 19:00, 19:10) \rangle, \langle 36.6, 36.9, 37.1, \emptyset, \emptyset, \emptyset \rangle, \langle 75, 78, 80, \emptyset, \emptyset, \emptyset \rangle \rrbracket$$

Далі потрібно відсортувати отриманий мультиобраз  $G_1$  за часовим кортежем за допомогою відповідної операції АСА:

$$G_2 = G_1 \uparrow \langle \tilde{t} \rangle_{i=1}^3 = \llbracket \tilde{T}, M_t, M_p \mid \langle (8:30, 9:00, 9:30), (8:45, 9:00, 9:15), (13:30, 14:00, 14:30), (13:45, 14:00, 14:15), (18:45, 19:00, 19:15), (18:50, 19:00, 19:10) \rangle, \langle 36.6, \emptyset, 36.9, \emptyset, 37.1, \emptyset \rangle, \langle 75, \emptyset, 78, \emptyset, 80, \emptyset \rangle \rrbracket$$

Тепер кортежі впорядковані, проте вони включають деякі зайві елементи, такі, як дублікати нечітких значень часу та фіктивні елементи. Щоб видалити їх, застосуємо операцію проріджування:

$$G_3 = G_2 \parallel \langle \tilde{t} \rangle_{i=1}^3 = \llbracket \tilde{T}, M_t, M_p \mid \langle (8:30, 9:00, 9:30), (13:30, 14:00, 14:30), (18:45, 19:00, 19:15) \rangle, \langle 36.6, 36.9, 37.1 \rangle, \langle 75, 78, 80 \rangle \rrbracket$$

В результаті отримуємо мультиобраз, який містить дані різних обстежень пацієнта та може бути використаний для аналізу цих даних.

При вирішенні деяких задач аналізу темпоральних мультимодальних даних може виникати потреба у синхронізації декількох мультиобразів.

Розглянемо метод, що дозволяє виконати таку синхронізацію, у тому числі у випадку, коли значення часу є нечітко визначеними.

### 3.3. Метод синхронізації мультиобразів

Метод синхронізації мультиобразів ґрунтується на застосуванні інтервальних відношень (2.89)-(2.90) та виконанні правил синхронізації.

Правила синхронізації поділяються на:

- універсальне правило синхронізації (УП),
- базові правила синхронізації (БП),
- правила нечіткої синхронізації (НП).

Універсальне правило синхронізації дозволяє виконати точну синхронізацію темпоральних мультимодальних даних та може бути застосоване до довільних кортежів значень часу, проте воно потребує виконання операцій, кількість яких не може бути зменшена.

Базові правила синхронізації призначені для виконання точної синхронізації, причому якщо вхідні кортежі значень часу є періодичними послідовностями, то кількість операцій над даними може бути зменшена за рахунок застосування відповідного *шаблону синхронізації* – одного чи декількох правил синхронізації, які відповідають виду періодичності у часових кортежах, що синхронізуються. Таким чином, застосування шаблону синхронізації, що створений шляхом комбінування базових правил синхронізації, дозволяє підвищити ефективність обробки даних.

Правила нечіткої синхронізації дозволяють виконати синхронізацію з деякою припустимою похибкою визначення елементів кортежів значень часу у мультиобразах, що синхронізуються; вони також можуть бути застосовані для створення шаблону синхронізації з метою підвищення ефективності обробки даних мультиобразів.

Сформулюємо правила всіх типів для синхронізації двох мультиобразів, заданих виразом (3.20):

$$\begin{aligned} I_1 &= \llbracket T, M_1 | \langle t_{i_1}^1 \rangle, \langle a_{i_1}^1 \rangle \rrbracket_{i_1=1}^{n_1} \\ I_2 &= \llbracket T, M_2 | \langle t_{i_2}^2 \rangle, \langle a_{i_2}^2 \rangle \rrbracket_{i_2=1}^{n_2} \end{aligned} \quad (3.20)$$

В результаті застосування деякого правила синхронізації (шаблону синхронізації) отримаємо два мультиобрази, задані виразом (3.21):

$$\begin{aligned} I_1^S &= \llbracket T, M_1 | \langle t_i \rangle, \langle d_i^1 \rangle \rrbracket_{i=1}^n \\ I_2^S &= \llbracket T, M_2 | \langle t_i \rangle, \langle d_i^2 \rangle \rrbracket_{i=1}^n, \end{aligned} \quad (3.21)$$

де  $d_i^1$  та  $d_i^2$  – значення, що належать нормалізованим кортежам елементів мультиобразів  $I_1$  та  $I_2$  відповідно;  $n = n_1 + n_2 - \delta$ ,  $\delta$  – кількість дублікатів значень часу у об'єднаному кортежі  $\langle t_i \rangle_{i=1}^n$  часових значень.

Універсальне правило синхронізації УП1 мультиобразів (3.20) задається виразами (3.22):

$$\begin{aligned} \langle t_i \rangle_{i=1}^{n_1+n_2-\delta} &= \left( \left( \left( \langle t_{i_1}^1 \rangle_{i_1=1}^{n_1} \cup \langle t_{i_2}^2 \rangle_{i_2=1}^{n_2} \right) \uparrow \right) \parallel \right) \\ d_i^1 &= \begin{cases} a_{i_1}^1 & \text{якщо } t_i = t_{i_1}^1, \forall i, \forall i_1 \\ \emptyset & \text{в інших випадках} \end{cases}, \\ d_i^2 &= \begin{cases} a_{i_2}^2 & \text{якщо } t_i = t_{i_2}^2, \forall i, \forall i_2 \\ \emptyset & \text{в інших випадках} \end{cases} \end{aligned} \quad (3.22)$$

де  $\cup$  – операція об'єднання,  $\uparrow$  – операція сортування,  $\parallel$  – операція проріджування.

Визначимо тепер базові правила синхронізації.

### 3.3.1. Базові правила синхронізації

Розглянемо можливі інтервальні відношення між дискретними інтервалами  $\bar{t}^1 = \langle t_i^1 \rangle_{i=1}^{n_1}$  та  $\bar{t}^2 = \langle t_i^2 \rangle_{i=1}^{n_2}$  та сформулюємо для кожного з цих відношень базові правила синхронізації.

Нехай  $e_d(\bar{t}^1, \bar{t}^2) = true$ , тобто  $\bar{t}^1$  та  $\bar{t}^2$  зв'язані відношенням *збігається з*, що визначене (2.58). З практичної точки зору це означає, що визначення (вимірювання, генерування) значень обох кортежів  $\bar{a}^1$  та  $\bar{a}^2$  відбувалось одночасно.

Можливі випадки для відношення *збігається з* проілюстровані графічно на рис. 3.3-3.5.

Перша група випадків для відношення  $e_d(\bar{t}^1, \bar{t}^2)$  характеризується тим, що  $n_1 \bmod 2 = 0$  та  $\bar{t}^1 \sim \bar{t}^2$ . Останнє означає, що  $|\bar{t}^1| = |\bar{t}^2|$  (тобто  $n_1 = n_2$ ).

Якщо  $t_i^1 = t_i^2$ ,  $\forall i \in [1 \dots n_1]$ , то  $n = n_1$  та значення елементів мультиобразів  $I_1^S$  та  $I_2^S$ , що визначені (3.21), отримуються згідно з базовим правилом синхронізації БП1, що визначається виразами (3.23) та може бути графічно проілюстроване рис. 3.3а.

$$\begin{aligned} t_i &= t_i^1 \\ d_i^1 &= a_i^1 \\ d_i^2 &= a_i^2 \end{aligned} \tag{3.23}$$

У випадку, коли  $t_i^1 = t_i^2$  та  $t_{i+1}^1 > t_{i+1}^2$ , що проілюстровано на рис. 3.3б, значення всіх елементів мультиобразів  $I_1^S$  та  $I_2^S$ , крім перших та останніх, отримуються згідно з виразами (3.24).

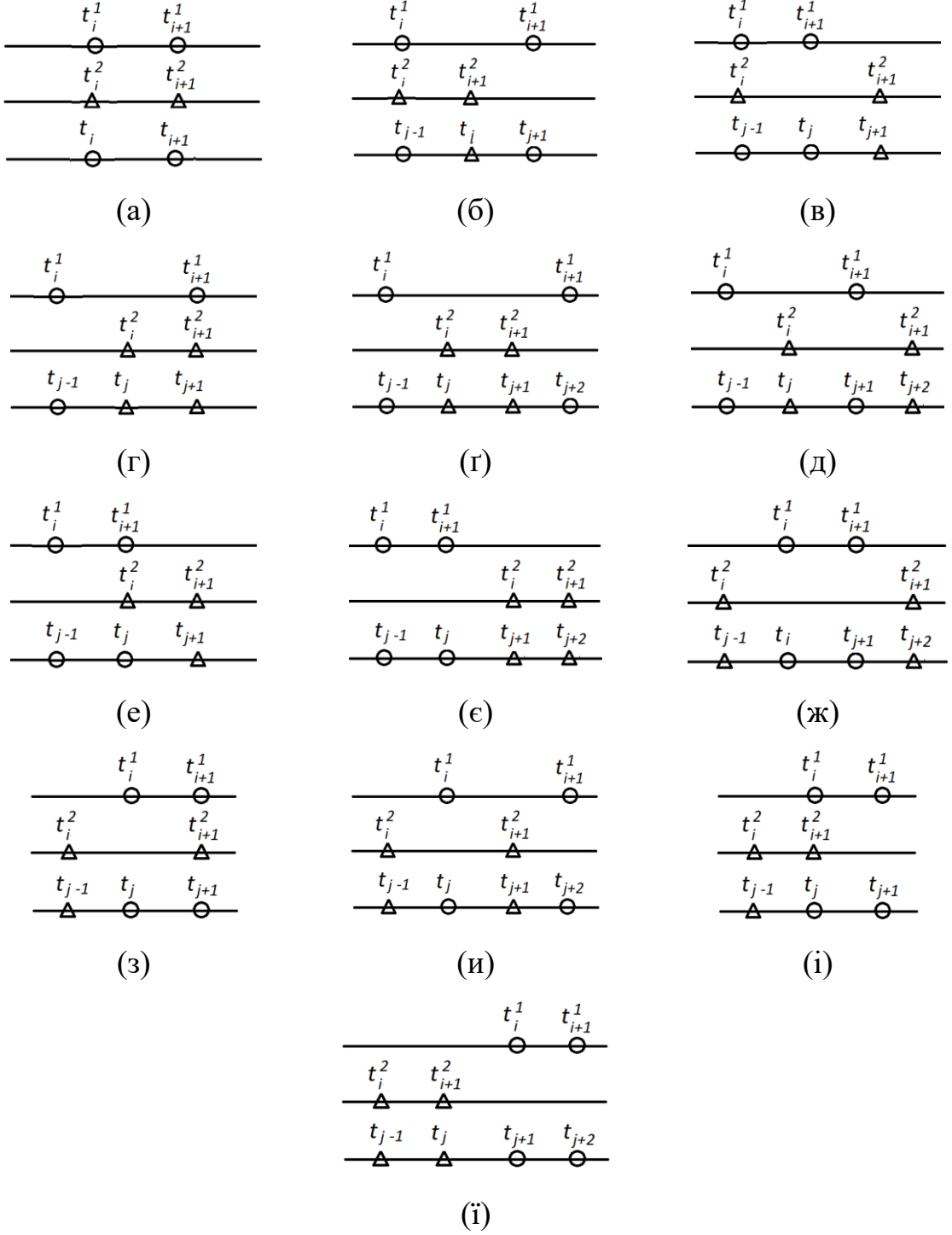


Рис. 3.3. Випадки для  $e_d(\bar{t}^1, \bar{t}^2)$  при  $\bar{t}^1 \sim \bar{t}^2$

$$\begin{array}{lll}
 t_{j-1} = t_i^1 & t_j = t_{i+1}^2 & t_{j+1} = t_{i+1}^1 \\
 d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 \\
 d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset
 \end{array} \quad (3.24)$$

Це означає, що у момент часу  $t_i^1 = t_i^2$  визначеними є обидва значення  $a_i^1$  та  $a_i^2$ ; у момент часу  $t_{i+1}^2$  визначеним є лише значення  $a_{i+1}^2$ ; у момент часу  $t_{i+1}^1$  визначеним є лише значення  $a_{i+1}^1$ . Індекс  $j$  обчислюється як  $j = \frac{3i}{2}$ , де  $i \in [2 \dots (n_1 - 2)]$  та  $i \bmod 2 = 0$ . Загальна кількість елементів в кожному спільному кортежі, а саме, у кортежах  $\langle t_j \rangle_{j=1}^n$ ,  $\langle d_j^1 \rangle_{j=1}^n$ ,  $\langle d_j^2 \rangle_{j=1}^n$ , складає  $n = \frac{3n_1}{2} - 1$ . Таким чином, правило синхронізації мультимодальних даних БП2 визначається виразами (3.25):

$$\begin{array}{ccccc} t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_{i+1}^2 & t_{j+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\ d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\ d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & d_n^2 = a_{n_2}^2 \end{array} \quad (3.25)$$

Аналогічно, якщо  $t_i^1 = t_i^2$  та  $t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3в),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ ,  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$ , то синхронізація значень кортежів визначається правилом БП3, що визначається виразами (3.26):

$$\begin{array}{ccccc} t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 & t_n = t_{n_1}^1 \\ d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\ d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 & d_n^2 = a_{n_2}^2 \end{array} \quad (3.26)$$

Якщо  $t_i^1 < t_i^2$  та  $t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3г),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$  та правило синхронізації даних БП4:

$$\begin{array}{ccccc} t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\ d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\ d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 & d_n^2 = a_{n_2}^2 \end{array} \quad (3.27)$$

Якщо  $t_i^1 < t_i^2$  та  $t_{i+1}^1 > t_{i+1}^2$  (рис. 3.3г),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації даних БП5:

$$\begin{aligned} t_1 &= t_1^1 & t_{j-2} &= t_i^1 & t_{j-1} &= t_i^2 & t_j &= t_{i+1}^2 & t_{j+1} &= t_{i+1}^1 & t_n &= t_{n_1}^1 \\ d_1^1 &= a_1^1 & d_{j-2}^1 &= a_i^1 & d_{j-1}^1 &= \emptyset & d_j^1 &= \emptyset & d_{j+1}^1 &= a_{i+1}^1 & d_n^1 &= a_{n_1}^1 \\ d_1^2 &= a_1^2 & d_{j-2}^2 &= \emptyset & d_{j-1}^2 &= a_i^2 & d_j^2 &= a_{i+1}^2 & d_{j+1}^2 &= \emptyset & d_n^2 &= a_{n_2}^2 \end{aligned} \quad (3.28)$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$  та  $t_{i+1}^1 > t_i^2$  (рис. 3.3д),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та синхронізація значень кортежів визначається правилом БП6, що визначається виразами (3.29).

$$\begin{aligned} t_1 &= t_1^1 & t_{j-2} &= t_i^1 & t_{j-1} &= t_i^2 & t_j &= t_{i+1}^1 & t_{j+1} &= t_{i+1}^2 & t_n &= t_{n_1}^1 \\ d_1^1 &= a_1^1 & d_{j-2}^1 &= a_i^1 & d_{j-1}^1 &= \emptyset & d_j^1 &= a_{i+1}^1 & d_{j+1}^1 &= \emptyset & d_n^1 &= a_{n_1}^1 \\ d_1^2 &= a_1^2 & d_{j-2}^2 &= \emptyset & d_{j-1}^2 &= a_i^2 & d_j^2 &= \emptyset & d_{j+1}^2 &= a_{i+1}^2 & d_n^2 &= a_{n_2}^2 \end{aligned} \quad (3.29)$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$  та  $t_{i+1}^1 = t_i^2$  (рис. 3.3е),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$  та правило синхронізації даних БП7:

$$\begin{aligned} t_1 &= t_1^1 & t_{j-1} &= t_i^1 & t_j &= t_i^2 & t_{j+1} &= t_{i+1}^2 & t_n &= t_{n_1}^1 \\ d_1^1 &= a_1^1 & d_{j-1}^1 &= a_i^1 & d_j^1 &= a_{i+1}^1 & d_{j+1}^1 &= \emptyset & d_n^1 &= a_{n_1}^1 \\ d_1^2 &= a_1^2 & d_{j-1}^2 &= \emptyset & d_j^2 &= a_i^2 & d_{j+1}^2 &= a_{i+1}^2 & d_n^2 &= a_{n_2}^2 \end{aligned} \quad (3.30)$$

Якщо  $t_{i+1}^1 < t_i^2$  (рис. 3.3є),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БП8:

$$\begin{array}{l}
t_1 = t_1^1 \quad t_{j-2} = t_i^1 \quad t_{j-1} = t_{i+1}^1 \quad t_j = t_i^2 \quad t_{j+1} = t_{i+1}^2 \quad t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 \quad d_{j-2}^1 = a_i^1 \quad d_{j-1}^1 = a_{i+1}^1 \quad d_j^1 = \emptyset \quad d_{j+1}^1 = \emptyset \quad d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 \quad d_{j-2}^2 = \emptyset \quad d_{j-1}^2 = \emptyset \quad d_j^2 = a_i^2 \quad d_{j+1}^2 = a_{i+1}^2 \quad d_n^2 = a_{n_2}^2
\end{array} \quad (3.31)$$

Якщо  $t_i^1 > t_i^2$  та  $t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3ж),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації даних БП9:

$$\begin{array}{l}
t_1 = t_1^1 \quad t_{j-2} = t_i^2 \quad t_{j-1} = t_i^1 \quad t_j = t_{i+1}^1 \quad t_{j+1} = t_{i+1}^2 \quad t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 \quad d_{j-2}^1 = \emptyset \quad d_{j-1}^1 = a_i^1 \quad d_j^1 = a_{i+1}^1 \quad d_{j+1}^1 = \emptyset \quad d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 \quad d_{j-2}^2 = a_i^2 \quad d_{j-1}^2 = \emptyset \quad d_j^2 = \emptyset \quad d_{j+1}^2 = a_{i+1}^2 \quad d_n^2 = a_{n_2}^2
\end{array} \quad (3.32)$$

Якщо  $t_i^1 > t_i^2$  та  $t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3з),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$  та синхронізація значень кортежів визначається правилом БП10, що визначається виразами (3.33).

$$\begin{array}{l}
t_1 = t_1^1 \quad t_{j-1} = t_i^2 \quad t_j = t_i^1 \quad t_{j+1} = t_{i+1}^1 \quad t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 \quad d_{j-1}^1 = \emptyset \quad d_j^1 = a_i^1 \quad d_{j+1}^1 = a_{i+1}^1 \quad d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 \quad d_{j-1}^2 = a_i^2 \quad d_j^2 = \emptyset \quad d_{j+1}^2 = a_{i+1}^2 \quad d_n^2 = a_{n_2}^2
\end{array} \quad (3.33)$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 > t_{i+1}^2$  та  $t_i^1 < t_{i+1}^2$  (рис. 3.3и),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації даних БП11:

$$\begin{array}{l}
t_1 = t_1^1 \quad t_{j-2} = t_i^2 \quad t_{j-1} = t_i^1 \quad t_j = t_{i+1}^2 \quad t_{j+1} = t_{i+1}^1 \quad t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 \quad d_{j-2}^1 = \emptyset \quad d_{j-1}^1 = a_i^1 \quad d_j^1 = \emptyset \quad d_{j+1}^1 = a_{i+1}^1 \quad d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 \quad d_{j-2}^2 = a_i^2 \quad d_{j-1}^2 = \emptyset \quad d_j^2 = a_{i+1}^2 \quad d_{j+1}^2 = \emptyset \quad d_n^2 = a_{n_2}^2
\end{array} \quad (3.34)$$

Якщо  $t_i^1 = t_{i+1}^2$  (рис. 3.3і),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$  та правило синхронізації мультимодальних даних БП12:



$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array} \quad (3.35)$$

Якщо  $t_i^1 > t_{i+1}^2$  (рис. 3.3ї),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БП13:

$$\begin{array}{cccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_{i+1}^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = a_{i+1}^2 & d_j^2 = \emptyset & d_{j+1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array} \quad (3.36)$$

Друга група випадків для  $e_d(\bar{t}^1, \bar{t}^2)$  характеризується тим, що  $n_1 \bmod 2 \neq 0$  та  $\bar{t}^1 \sim \bar{t}^2$ . Часове узгодження  $i$ -х елементів та  $(i+1)$ -х елементів є подібним до узгодження для першої групи випадків (рис. 3.3), проте, оскільки  $n_1$  є непарним, то для синхронізації значень потрібно також аналізувати значення часу  $t_{n_1-1}^1$  та  $t_{n_2-1}^2$ .

Якщо  $t_i^1 = t_i^2$  (рис. 3.3а),  $\forall i \in [1 \dots n_1]$ , то синхронізація відбувається згідно з (3.23).

Якщо  $t_i^1 = t_i^2$ ,  $t_{i+1}^1 > t_{i+1}^2$  (рис. 3.3б),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2}$  та правило синхронізації даних БП14:

$$\begin{array}{cccccc}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_{i+1}^2 & t_{j+1} = t_{i+1}^1 & t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.37)$$

Якщо  $t_i^1 = t_i^2$ ,  $t_{i+1}^1 > t_{i+1}^2$  (рис. 3.3б),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та правило синхронізації БП15:

$$\begin{aligned}
t_1 &= t_1^1 & t_{j-1} &= t_i^1 & t_j &= t_{i+1}^2 & t_{j+1} &= t_{i+1}^1 & t_{n-2} &= t_{n_1-1}^1 \\
d_1^1 &= a_1^1 & d_{j-1}^1 &= a_i^1 & d_j^1 &= \emptyset & d_{j+1}^1 &= a_{i+1}^1 & d_{n-2}^1 &= a_{n_1-1}^1 \\
d_1^2 &= a_1^2 & d_{j-1}^2 &= a_i^2 & d_j^2 &= a_{i+1}^2 & d_{j+1}^2 &= \emptyset & d_{n-2}^2 &= \emptyset \\
t_{n-1} &= t_{n_2-1}^2 & & & & & & & t_n &= t_{n_1}^1 \\
d_{n-1}^1 &= \emptyset & & & & & & & d_n^1 &= a_{n_1}^1 \\
d_{n-1}^2 &= a_{n_2-1}^2 & & & & & & & d_n^2 &= a_{n_2}^2
\end{aligned} \tag{3.38}$$

Якщо  $t_i^1 = t_i^2, t_{i+1}^1 > t_{i+1}^2$  (рис. 3.3б),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та синхронізація значень кортежів визначається правилом БП16, що визначається виразами (3.39).

$$\begin{aligned}
t_1 &= t_1^1 & t_{j-1} &= t_i^1 & t_j &= t_{i+1}^2 & t_{j+1} &= t_{i+1}^1 & t_{n-2} &= t_{n_2-1}^2 \\
d_1^1 &= a_1^1 & d_{j-1}^1 &= a_i^1 & d_j^1 &= \emptyset & d_{j+1}^1 &= a_{i+1}^1 & d_{n-2}^1 &= \emptyset \\
d_1^2 &= a_1^2 & d_{j-1}^2 &= a_i^2 & d_j^2 &= a_{i+1}^2 & d_{j+1}^2 &= \emptyset & d_{n-2}^2 &= a_{n_2-1}^2 \\
t_{n-1} &= t_{n_1-1}^1 & & & & & & & t_n &= t_{n_1}^1 \\
d_{n-1}^1 &= a_{n_1-1}^1 & & & & & & & d_n^1 &= a_{n_1}^1 \\
d_{n-1}^2 &= \emptyset & & & & & & & d_n^2 &= a_{n_2}^2
\end{aligned} \tag{3.39}$$

Якщо  $t_i^1 = t_i^2, t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3в),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2}$  та правило синхронізації даних БП17:

$$\begin{aligned}
t_1 &= t_1^1 & t_{j-1} &= t_i^1 & t_j &= t_{i+1}^1 & t_{j+1} &= t_{i+1}^2 & t_{n-1} &= t_{n_1-1}^1 & t_n &= t_{n_1}^1 \\
d_1^1 &= a_1^1 & d_{j-1}^1 &= a_i^1 & d_j^1 &= a_{i+1}^1 & d_{j+1}^1 &= \emptyset & d_{n-1}^1 &= a_{n_1-1}^1 & d_n^1 &= a_{n_1}^1 \\
d_1^2 &= a_1^2 & d_{j-1}^2 &= a_i^2 & d_j^2 &= \emptyset & d_{j+1}^2 &= a_{i+1}^2 & d_{n-1}^2 &= a_{n_2-1}^2 & d_n^2 &= a_{n_2}^2
\end{aligned} \tag{3.40}$$

Якщо  $t_i^1 = t_i^2, t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3в),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та правило синхронізації БП18:

$$\begin{array}{llllll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 & t_{n-2} = t_{n_1-1}^1 & \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset & d_{n-2}^1 = a_{n_1-1}^1 & \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = \emptyset & \\
\end{array}
\quad (3.41)$$

$$\begin{array}{ll}
t_{n-1} = t_{n_2-1}^2 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 = t_i^2, t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3в),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та синхронізація значень кортежів визначається правилом БП19, що визначається виразами (3.42).

$$\begin{array}{llllll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 & t_{n-2} = t_{n_2-1}^2 & \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset & d_{n-2}^1 = \emptyset & \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = a_{n_2-1}^2 & \\
\end{array}
\quad (3.42)$$

$$\begin{array}{ll}
t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 < t_i^2, t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3г),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2}$  та правило синхронізації даних БП20:

$$\begin{array}{llllllll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^1 & t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 & & \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 & & \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 & d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2 & & 
\end{array}
\quad (3.43)$$

Якщо  $t_i^1 < t_i^2, t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3г),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та правило синхронізації БП21:

$$\begin{array}{lllll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^1 & t_{n-2} = t_{n_1-1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & d_{n-2}^1 = a_{n_1-1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = \emptyset
\end{array}
\tag{3.44}$$

$$\begin{array}{ll}
t_{n-1} = t_{n_2-1}^2 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3Г),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та синхронізація значень кортежів визначається правилом БП22, що визначається виразами (3.45).

$$\begin{array}{lllll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^1 & t_{n-2} = t_{n_2-1}^2 \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & d_{n-2}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = a_{n_2-1}^2
\end{array}
\tag{3.45}$$

$$\begin{array}{ll}
t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 > t_{i+1}^2$  (рис. 3.3Г),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$  then  $j = 2i$ ,  $n = 2n_1 - 3$  та правило синхронізації БП23:

$$\begin{array}{lllll}
t_1 = t_1^1 & t_{j-2} = t_i^1 & t_{j-1} = t_i^2 & t_j = t_{i+1}^2 & t_{j+1} = t_{i+1}^1 \\
d_1^1 = a_1^1 & d_{j-2}^1 = a_i^1 & d_{j-1}^1 = \emptyset & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 \\
d_1^2 = a_1^2 & d_{j-2}^2 = \emptyset & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset
\end{array}
\tag{3.46}$$

$$\begin{array}{ll}
t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 < t_i^2, t_{i+1}^1 > t_{i+1}^2$  (рис. 3.3г),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$   
та  $t_{n_1-1}^1 < t_{n_2-1}^2$  then  $j = 2i, n = 2(n_1 - 1)$  та правило синхронізації БП24:

$$\begin{aligned}
 t_1 &= t_1^1 & t_{j-2} &= t_i^1 & t_{j-1} &= t_i^2 & t_j &= t_{i+1}^2 & t_{j+1} &= t_{i+1}^1 & t_{n-2} &= t_{n_1-1}^1 \\
 d_1^1 &= a_1^1 & d_{j-2}^1 &= a_i^1 & d_{j-1}^1 &= \emptyset & d_j^1 &= \emptyset & d_{j+1}^1 &= a_{i+1}^1 & d_{n-2}^1 &= a_{n_1-1}^1 \\
 d_1^2 &= a_1^2 & d_{j-2}^2 &= \emptyset & d_{j-1}^2 &= a_i^2 & d_j^2 &= a_{i+1}^2 & d_{j+1}^2 &= \emptyset & d_{n-2}^2 &= \emptyset \\
 t_{n-1} &= t_{n_2-1}^2 & & & & & & & & & t_n &= t_{n_1}^1 \\
 d_{n-1}^1 &= \emptyset & & & & & & & & & d_n^1 &= a_{n_1}^1 \\
 d_{n-1}^2 &= a_{n_2-1}^2 & & & & & & & & & d_n^2 &= a_{n_2}^2
 \end{aligned} \tag{3.47}$$

Якщо  $t_i^1 < t_i^2, t_{i+1}^1 > t_{i+1}^2$  (рис. 3.3г),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$   
та  $t_{n_1-1}^1 > t_{n_2-1}^2$  then  $j = 2i, n = 2(n_1 - 1)$  та правило синхронізації БП25:

$$\begin{aligned}
 t_1 &= t_1^1 & t_{j-2} &= t_i^1 & t_{j-1} &= t_i^2 & t_j &= t_{i+1}^2 & t_{j+1} &= t_{i+1}^1 \\
 d_1^1 &= a_1^1 & d_{j-2}^1 &= a_i^1 & d_{j-1}^1 &= \emptyset & d_j^1 &= \emptyset & d_{j+1}^1 &= a_{i+1}^1 \\
 d_1^2 &= a_1^2 & d_{j-2}^2 &= \emptyset & d_{j-1}^2 &= a_i^2 & d_j^2 &= a_{i+1}^2 & d_{j+1}^2 &= \emptyset \\
 t_{n-2} &= t_{n_2-}^2 & & & t_{n-1} &= t_{n_1}^1 & & & t_n &= t_{n_1}^1 \\
 d_{n-2}^1 &= \emptyset & & & d_{n-1}^1 &= a_{n_1}^1 & & & d_n^1 &= a_{n_1}^1 \\
 d_{n-2}^2 &= a_{n_2}^2 & & & d_{n-1}^2 &= \emptyset & & & d_n^2 &= a_{n_2}^2
 \end{aligned} \tag{3.48}$$

Якщо  $t_i^1 < t_i^2, t_{i+1}^1 < t_{i+1}^2, t_{i+1}^1 > t_i^2$  (рис. 3.3д),  $\forall i \in [2 \dots (n_1 - 2)], i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = 2i, n = 2n_1 - 3$  та правило синхронізації мультимодальних даних БП26:

$$\begin{aligned}
 t_1 &= t_1^1 & t_{j-2} &= t_i^1 & t_{j-1} &= t_i^2 & t_j &= t_{i+1}^1 & t_{j+1} &= t_{i+1}^2 \\
 d_1^1 &= a_1^1 & d_{j-2}^1 &= a_i^1 & d_{j-1}^1 &= \emptyset & d_j^1 &= a_{i+1}^1 & d_{j+1}^1 &= \emptyset \\
 d_1^2 &= a_1^2 & d_{j-2}^2 &= \emptyset & d_{j-1}^2 &= a_i^2 & d_j^2 &= \emptyset & d_{j+1}^2 &= a_{i+1}^2
 \end{aligned} \tag{3.49}$$

$$\begin{array}{ll}
t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$ ,  $t_{i+1}^1 > t_i^2$  (рис. 3.3д),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БП27:

$$\begin{array}{lllll}
t_1 = t_1^1 & t_{j-2} = t_i^1 & t_{j-1} = t_i^2 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = a_i^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = \emptyset & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 \\
t_{n-2} = t_{n_1-1}^1 & t_{n-1} = t_{n_2-1}^2 & & t_n = t_{n_1}^1 & \\
d_{n-2}^1 = a_{n_1-1}^1 & d_{n-1}^1 = \emptyset & & d_n^1 = a_{n_1}^1 & \\
d_{n-2}^2 = \emptyset & d_{n-1}^2 = a_{n_2-1}^2 & & d_n^2 = a_{n_2}^2 & 
\end{array} \quad (3.50)$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$ ,  $t_{i+1}^1 > t_i^2$  (рис. 3.3д),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних ПБ28:

$$\begin{array}{lllll}
t_1 = t_1^1 & t_{j-2} = t_i^1 & t_{j-1} = t_i^2 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = a_i^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = \emptyset & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 \\
t_{n-2} = t_{n_2-1}^2 & t_{n-1} = t_{n_1-1}^1 & & t_n = t_{n_1}^1 & \\
d_{n-2}^1 = \emptyset & d_{n-1}^1 = a_{n_1-1}^1 & & d_n^1 = a_{n_1}^1 & \\
d_{n-2}^2 = a_{n_2-1}^2 & d_{n-1}^2 = \emptyset & & d_n^2 = a_{n_2}^2 & 
\end{array} \quad (3.51)$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$ ,  $t_{i+1}^1 = t_i^2$  (рис. 3.3е),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2}$  та правило синхронізації мультимодальних даних БП29:

$$\begin{array}{llll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 \\
\\ 
t_{n-1} = t_{n_1-1}^1 & & t_n = t_{n_1}^1 & \\
d_{n-1}^1 = a_{n_1-1}^1 & & d_n^1 = a_{n_1}^1 & \\
d_{n-1}^2 = a_{n_2-1}^2 & & d_n^2 = a_{n_2}^2 & 
\end{array} \quad (3.52)$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$ ,  $t_{i+1}^1 = t_i^2$  (рис. 3.3е),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та синхронізація значень кортежів визначається правилом БП30, що визначається виразами (3.53).

$$\begin{array}{llllll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^2 & t_{n-2} = t_{n_1-1}^1 & \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset & d_{n-2}^1 = a_{n_1-1}^1 & \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = \emptyset & \\
\\ 
t_{n-1} = t_{n_2-1}^2 & & t_n = t_{n_1}^1 & & & \\
d_{n-1}^1 = \emptyset & & d_n^1 = a_{n_1}^1 & & & \\
d_{n-1}^2 = a_{n_2-1}^2 & & d_n^2 = a_{n_2}^2 & & & 
\end{array} \quad (3.53)$$

Якщо  $t_i^1 < t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$ ,  $t_{i+1}^1 = t_i^2$  (рис. 3.3е),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та правило синхронізації мультимодальних даних БП31:

$$\begin{array}{llllll}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^2 & t_{n-2} = t_{n_2-1}^2 & \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset & d_{n-2}^1 = \emptyset & \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = a_{n_2-1}^2 & 
\end{array} \quad (3.54)$$

$$\begin{array}{ll}
t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_{i+1}^1 < t_i^2$  (рис. 3.3є),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2n_1 - 3$  та правило синхронізації мультимодальних даних БПЗ2:

$$\begin{array}{lllll}
t_1 = t_1^1 & t_{j-2} = t_i^1 & t_{j-1} = t_{i+1}^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = a_i^1 & d_{j-1}^1 = a_{i+1}^1 & d_j^1 = \emptyset & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = \emptyset & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2
\end{array} \tag{3.55}$$

$$\begin{array}{ll}
t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_{i+1}^1 < t_i^2$  (рис. 3.3є),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БПЗ3:

$$\begin{array}{lllll}
t_1 = t_1^1 & t_{j-2} = t_i^1 & t_{j-1} = t_{i+1}^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = a_i^1 & d_{j-1}^1 = a_{i+1}^1 & d_j^1 = \emptyset & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = \emptyset & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2
\end{array} \tag{3.56}$$

$$\begin{array}{lll}
t_{n-2} = t_{n_1-1}^1 & t_{n-1} = t_{n_2-1}^2 & t_n = t_{n_1}^1 \\
d_{n-2}^1 = a_{n_1-1}^1 & d_{n-1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = \emptyset & d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_{i+1}^1 < t_i^2$  (рис. 3.3є),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БПЗ4:



$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^1 & t_{j-1} = t_{i+1}^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = a_i^1 & d_{j-1}^1 = a_{i+1}^1 & d_j^1 = \emptyset & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = \emptyset & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2
\end{array} \quad (3.57)$$

$$\begin{array}{ccc}
t_{n-2} = t_{n_2-1}^2 & t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-2}^1 = \emptyset & d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = a_{n_2-1}^2 & d_{n-1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3ж),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2n_1 - 3$  та синхронізація значень кортежів визначається правилом БП35, що визначається виразами (3.58).

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_i^1 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = \emptyset & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2
\end{array} \quad (3.58)$$

$$\begin{array}{ccc}
t_{n-1} = t_{n_1-1}^1 & & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3ж),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило БП36:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_i^1 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = \emptyset & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2
\end{array} \quad (3.59)$$

$$\begin{array}{ccc}
t_{n-2} = t_{n_1-1}^1 & t_{n-1} = t_{n_2-1}^2 & t_n = t_{n_1}^1 \\
d_{n-2}^1 = a_{n_1-1}^1 & d_{n-1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = \emptyset & d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 < t_{i+1}^2$  (рис. 3.3ж),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  
 $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило БП37:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_i^1 & t_j = t_{i+1}^1 & t_{j+1} = t_{i+1}^2 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = \emptyset & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 \\
t_{n-2} = t_{n_2-1}^2 & & t_{n-1} = t_{n_1-1}^1 & & t_n = t_{n_1}^1 \\
d_{n-2}^1 = \emptyset & & d_{n-1}^1 = a_{n_1-1}^1 & & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = a_{n_2-1}^2 & & d_{n-1}^2 = \emptyset & & d_n^2 = a_{n_2}^2
\end{array} \quad (3.60)$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3з),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  
 $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2}$  та правило БП38:

$$\begin{array}{cccccc}
t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 & d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.61)$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3з),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  
 $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та правило БП39:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_{n-2} = t_{n_1-1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_{n-2}^1 = a_{n_1-1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = \emptyset \\
t_{n-1} = t_{n_2-1}^2 & & & & t_n = t_{n_1}^1 \\
d_{n-1}^1 = \emptyset & & & & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & & & & d_n^2 = a_{n_2}^2
\end{array} \quad (3.62)$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 = t_{i+1}^2$  (рис. 3.3з),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  
 $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та правило БП40:

$$\begin{array}{llllll}
 t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_{n-2} = t_{n_2-1}^2 & \\
 d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_{n-2}^1 = \emptyset & \\
 d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 & d_{n-2}^2 = a_{n_2-1}^2 & \\
 t_{n-1} = t_{n_1-1}^1 & & & & t_n = t_{n_1}^1 & \\
 d_{n-1}^1 = a_{n_1-1}^1 & & & & d_n^1 = a_{n_1}^1 & \\
 d_{n-1}^2 = \emptyset & & & & d_n^2 = a_{n_2}^2 & 
 \end{array} \quad (3.63)$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 > t_{i+1}^2$ ,  $t_i^1 < t_{i+1}^2$  (рис. 3.3и),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  
 $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2n_1 - 3$  та синхронізація  
значень кортежів визначається правилом БП41, що визначається  
виразами (3.64).

$$\begin{array}{llllll}
 t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_i^1 & t_j = t_{i+1}^2 & t_{j+1} = t_{i+1}^1 & \\
 d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & \\
 d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & \\
 t_{n-1} = t_{n_1-1}^1 & & & & t_n = t_{n_1}^1 & \\
 d_{n-1}^1 = a_{n_1-1}^1 & & & & d_n^1 = a_{n_1}^1 & \\
 d_{n-1}^2 = a_{n_2-1}^2 & & & & d_n^2 = a_{n_2}^2 & 
 \end{array} \quad (3.64)$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 > t_{i+1}^2$ ,  $t_i^1 < t_{i+1}^2$  (рис. 3.3и),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  
 $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило  
синхронізації мультимодальних даних БП42:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_i^1 & t_j = t_{i+1}^2 & t_{j+1} = t_{i+1}^1 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset
\end{array} \quad (3.65)$$

$$\begin{array}{ccc}
t_{n-2} = t_{n_1-1}^1 & t_{n-1} = t_{n_2-1}^2 & t_n = t_{n_1}^1 \\
d_{n-2}^1 = a_{n_1-1}^1 & d_{n-1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = \emptyset & d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 > t_i^2$ ,  $t_{i+1}^1 > t_{i+1}^2$ ,  $t_i^1 < t_{i+1}^2$  (рис. 3.3и),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БП43:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_i^1 & t_j = t_{i+1}^2 & t_{j+1} = t_{i+1}^1 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset
\end{array} \quad (3.66)$$

$$\begin{array}{ccc}
t_{n-2} = t_{n_2-1}^2 & t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-2}^1 = \emptyset & d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = a_{n_2-1}^2 & d_{n-1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 = t_{i+1}^2$  (рис. 3.3і),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2}$  та правило синхронізації мультимодальних даних БП44:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_{n-1} = t_{n_1-1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_{n-1}^1 = a_{n_1-1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & d_{n-1}^2 = a_{n_2-1}^2
\end{array} \quad (3.67)$$

$$\begin{array}{c}
t_n = t_{n_1}^1 \\
d_n^1 = a_{n_1}^1 \\
d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 = t_{i+1}^2$  (рис. 3.3i),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та правило синхронізації мультимодальних даних БП45:

$$\begin{array}{ccccc}
 t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_{n-2} = t_{n_1-1}^1 \\
 d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_{n-2}^1 = a_{n_1-1}^1 \\
 d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & d_{n-2}^2 = \emptyset
 \end{array} \quad (3.68)$$

$$\begin{array}{cc}
 t_{n-1} = t_{n_2-1}^2 & t_n = t_{n_1}^1 \\
 d_{n-1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
 d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
 \end{array}$$

Якщо  $t_i^1 = t_{i+1}^2$  (рис. 3.3i),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3(n_1-1)}{2} + 1$  та синхронізація значень кортежів визначається правилом БП46, що визначається виразами (3.69).

$$\begin{array}{ccccc}
 t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_{n-2} = t_{n_2-1}^2 \\
 d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_{n-2}^1 = \emptyset \\
 d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & d_{n-2}^2 = a_{n_2-1}^2
 \end{array} \quad (3.69)$$

$$\begin{array}{cc}
 t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
 d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
 d_{n-1}^2 = \emptyset & d_n^2 = a_{n_2}^2
 \end{array}$$

Якщо  $t_i^1 > t_{i+1}^2$  (рис. 3.3ii),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 = t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2n_1 - 3$  та правило синхронізації мультимодальних даних БП47:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_{i+1}^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = a_{i+1}^2 & d_j^2 = \emptyset & d_{j+1}^2 = \emptyset
\end{array} \quad (3.70)$$

$$\begin{array}{ccc}
t_{n-1} = t_{n_1-1}^1 & & t_n = t_{n_1}^1 \\
d_{n-1}^1 = a_{n_1-1}^1 & & d_n^1 = a_{n_1}^1 \\
d_{n-1}^2 = a_{n_2-1}^2 & & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 > t_{i+1}^2$  (рис. 3.3ї),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 < t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БП48:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_{i+1}^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = a_{i+1}^2 & d_j^2 = \emptyset & d_{j+1}^2 = \emptyset
\end{array} \quad (3.71)$$

$$\begin{array}{ccc}
t_{n-2} = t_{n_1-1}^1 & t_{n-1} = t_{n_2-1}^2 & t_n = t_{n_1}^1 \\
d_{n-2}^1 = a_{n_1-1}^1 & d_{n-1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = \emptyset & d_{n-1}^2 = a_{n_2-1}^2 & d_n^2 = a_{n_2}^2
\end{array}$$

Якщо  $t_i^1 > t_{i+1}^2$  (рис. 3.3ї),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$  та  $t_{n_1-1}^1 > t_{n_2-1}^2$ , то  $j = 2i$ ,  $n = 2(n_1 - 1)$  та правило синхронізації мультимодальних даних БП49:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-2} = t_i^2 & t_{j-1} = t_{i+1}^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 \\
d_1^1 = a_1^1 & d_{j-2}^1 = \emptyset & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 \\
d_1^2 = a_1^2 & d_{j-2}^2 = a_i^2 & d_{j-1}^2 = a_{i+1}^2 & d_j^2 = \emptyset & d_{j+1}^2 = \emptyset
\end{array} \quad (3.72)$$

$$\begin{array}{ccc}
t_{n-2} = t_{n_2-1}^2 & t_{n-1} = t_{n_1-1}^1 & t_n = t_{n_1}^1 \\
d_{n-2}^1 = \emptyset & d_{n-1}^1 = a_{n_1-1}^1 & d_n^1 = a_{n_1}^1 \\
d_{n-2}^2 = a_{n_2-1}^2 & d_{n-1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array}$$

Третя група випадків для  $e_d(\bar{t}^1, \bar{t}^2)$  відрізняється тим, що  $\bar{t}^1 \triangleleft \bar{t}^2$ ,  $|\bar{t}^1| < |\bar{t}^2|$ . Оскільки коефіцієнт  $R$  для  $|\bar{t}^1|$  та  $|\bar{t}^2|$  може бути довільним, припустимо, що для кожному  $i$ -го елементи в  $\bar{t}^1$  відповідають два елементи в  $\bar{t}^2$ , за виключенням  $t_1^1$  та  $t_{n_1}^1$  що збігаються з  $t_1^2$  та  $t_{n_2}^2$  відповідно (рис. 3.4). Це означає, що  $n_2 \bmod 2 = 0$  та  $R = \frac{n_1}{2(n_1-1)}$ .

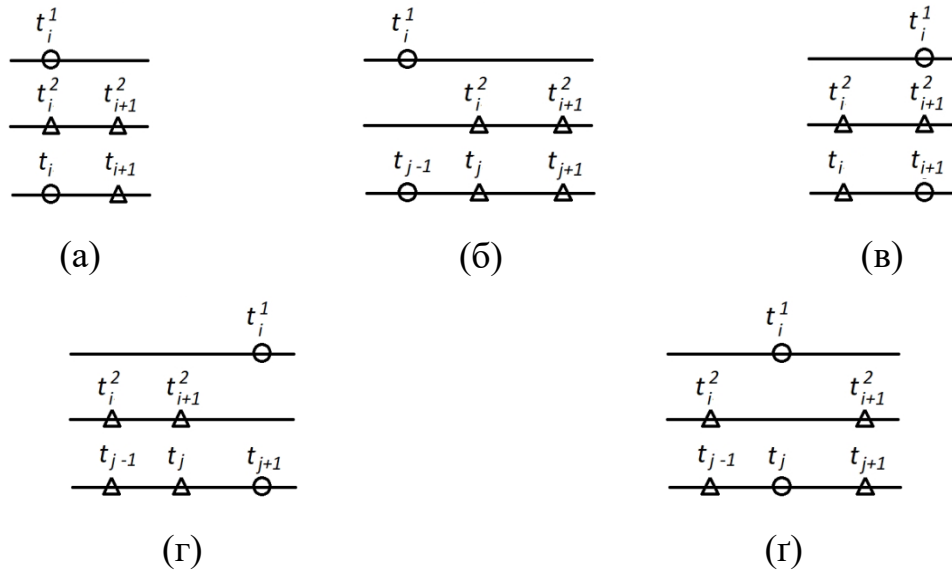


Рис. 3.4. Випадки для  $e_d(\bar{t}^1, \bar{t}^2)$  при  $\bar{t}^1 \triangleleft \bar{t}^2$

Якщо  $t_i^1 = t_i^2$  (рис. 3.4а),  $\forall i \in [2 \dots (n_2 - 2)]$ ,  $i \bmod 2 = 0$ , то  $n = n_2$  та правило синхронізації мультимодальних даних БП50:

$$\begin{array}{llll}
 t_1 = t_1^1 & t_i = t_i^1 & t_{i+1} = t_{i+1}^2 & t_n = t_{n_2}^1 \\
 d_1^1 = a_1^1 & d_i^1 = a_i^1 & d_{i+1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
 d_1^2 = a_1^2 & d_i^2 = a_i^2 & d_{i+1}^2 = a_{i+1}^2 & d_n^2 = a_{n_2}^2
 \end{array} \quad (3.73)$$

Якщо  $t_i^1 < t_i^2$  (рис. 3.4б),  $\forall i \in [2 \dots (n_2 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_2}{2} - 1$  та правило синхронізації мультимодальних даних БП51:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^2 & t_n = t_{n_2}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = a_{i+1}^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.74)$$

Якщо  $t_i^1 = t_{i+1}^2$  (рис. 3.4в),  $\forall i \in [2 \dots (n_2 - 2)]$ ,  $i \bmod 2 = 0$ , то  $n = n_2$  та правило синхронізації мультимодальних даних БП52:

$$\begin{array}{cccc}
t_1 = t_1^1 & t_i = t_i^2 & t_{i+1} = t_i^1 & t_n = t_{n_2}^1 \\
d_1^1 = a_1^1 & d_i^1 = \emptyset & d_{i+1}^1 = a_i^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_i^2 = a_i^2 & d_{i+1}^2 = a_{i+1}^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.75)$$

Якщо  $t_i^1 > t_{i+1}^2$  (рис. 3.4г),  $\forall i \in [2 \dots (n_2 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_2}{2} - 1$  та правило синхронізації мультимодальних даних БП53:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_{i+1}^2 & t_{j+1} = t_i^1 & t_n = t_{n_2}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = \emptyset & d_{j+1}^1 = a_i^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = a_{i+1}^2 & d_{j+1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array} \quad (3.76)$$

Якщо  $t_i^1 > t_i^2$  та  $t_i^1 < t_{i+1}^2$  (рис. 3.4г),  $\forall i \in [2 \dots (n_2 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_2}{2} - 1$  та правило синхронізації даних БП54:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^2 & t_n = t_{n_2}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = a_{i+1}^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.77)$$

Четверта група випадків для  $e_d(\bar{t}^1, \bar{t}^2)$  відрізняється тим, що  $\bar{t}^1 \triangleright \bar{t}^2$ , тобто  $|\bar{t}^1| > |\bar{t}^2|$ . Подібно випадкам третьої групи ( $\bar{t}^1 \triangleleft \bar{t}^2$ ), коефіцієнт  $R$  для  $|\bar{t}^1|$  та  $|\bar{t}^2|$  може бути довільним. Припустимо, що кожному  $i$ -му



елементу в  $\bar{t}^2$  відповідають два елементи в  $\bar{t}^1$ , за виключенням  $t_1^2$  та  $t_{n_2}^2$ , які збігаються з  $t_1^1$  та  $t_{n_1}^1$  відповідно (рис. 3.5). Тобто  $n_1 \bmod 2 = 0$  та  $R = \frac{n_2}{2(n_2-1)}$ .

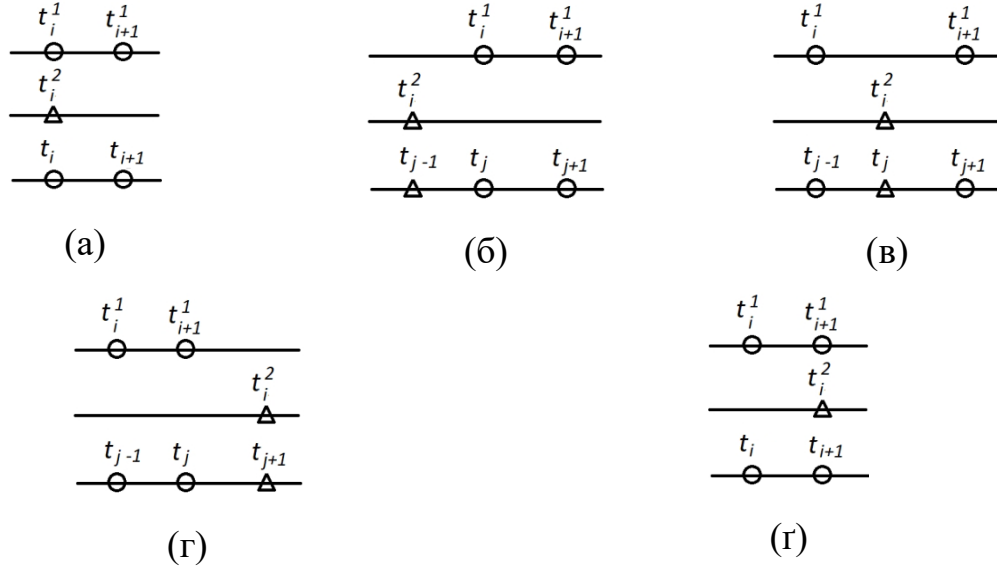


Рис. 3.5. Випадки для  $e_d(\bar{t}^1, \bar{t}^2)$  при  $\bar{t}^1 \triangleright \bar{t}^2$

Якщо  $t_i^1 = t_i^2$  (рис. 3.5а),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $n = n_1$  та правило синхронізації мультимодальних даних БП55:

$$\begin{array}{llll}
 t_1 = t_1^1 & t_i = t_i^1 & t_{i+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\
 d_1^1 = a_1^1 & d_i^1 = a_i^1 & d_{i+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\
 d_1^2 = a_1^2 & d_i^2 = a_i^2 & d_{i+1}^2 = \emptyset & d_n^2 = a_{n_2}^2
 \end{array} \quad (3.78)$$

Якщо  $t_i^1 > t_i^2$  (рис. 3.5б),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$  та правило синхронізації мультимодальних даних БП56:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^2 & t_j = t_i^1 & t_{j+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = \emptyset & d_j^1 = a_i^1 & d_{j+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = a_i^2 & d_j^2 = \emptyset & d_{j+1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array} \quad (3.79)$$

Якщо  $t_i^1 < t_i^2$  та  $t_{i+1}^1 > t_i^2$  (рис. 3.5в),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$  та правило синхронізації даних БП57:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_i^2 & t_{j+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = \emptyset & d_{j+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = a_i^2 & d_{j+1}^2 = \emptyset & d_n^2 = a_{n_2}^2
\end{array} \quad (3.80)$$

Якщо  $t_{i+1}^1 < t_i^2$  (рис. 3.5г),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $j = \frac{3i}{2}$ ,  $n = \frac{3n_1}{2} - 1$  та правило синхронізації мультимодальних даних БП58:

$$\begin{array}{ccccc}
t_1 = t_1^1 & t_{j-1} = t_i^1 & t_j = t_{i+1}^1 & t_{j+1} = t_i^2 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_{j-1}^1 = a_i^1 & d_j^1 = a_{i+1}^1 & d_{j+1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_{j-1}^2 = \emptyset & d_j^2 = \emptyset & d_{j+1}^2 = a_i^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.81)$$

Якщо  $t_{i+1}^1 = t_i^2$  (рис. 3.5г),  $\forall i \in [2 \dots (n_1 - 2)]$ ,  $i \bmod 2 = 0$ , то  $n = n_1$  та правило синхронізації мультимодальних даних БП59:

$$\begin{array}{cccc}
t_1 = t_1^1 & t_i = t_i^1 & t_{i+1} = t_{i+1}^1 & t_n = t_{n_1}^1 \\
d_1^1 = a_1^1 & d_i^1 = a_i^1 & d_{i+1}^1 = a_{i+1}^1 & d_n^1 = a_{n_1}^1 \\
d_1^2 = a_1^2 & d_i^2 = \emptyset & d_{i+1}^2 = a_i^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.82)$$

Більш загальний випадок синхронізації визначається відношенням відбувається між, що визначається згідно з (2.69). Він відрізняється тим, що отримання (вимірювань, генерація) однакової кількості значень

кортежів даних  $\bar{a}^1$  та  $\bar{a}^2$  відбувається у один і той самий період часу з моменту  $\alpha$  до моменту  $\beta$ , але не одночасно. Відмінність між відношенням *збігається з* та *відбувається між* полягає у способі визначення перших та останніх елементів кортежів мультиобразу, що.

Отже, сформулюємо правило синхронізації БП60, яке полягає у тому, що перші елементи кортежів визначаються відповідно до виразів (3.83), а останні елементи кортежів визначаються відповідно до виразів (3.84).

$$t_1 = \begin{cases} t_1^1 & \text{якщо } t_1^1 \leq t_1^2 \\ t_1^2 & \text{якщо } t_1^1 > t_1^2 \end{cases} \quad (3.83)$$

$$d_1^1 = \begin{cases} a_1^1 & \text{якщо } t_1^1 \leq t_1^2 \\ \emptyset & \text{якщо } t_1^1 > t_1^2 \end{cases} \quad d_1^2 = \begin{cases} \emptyset & \text{якщо } t_1^1 < t_1^2 \\ a_1^2 & \text{якщо } t_1^1 \geq t_1^2 \end{cases}$$

$$t_n = \begin{cases} t_{n_1}^1 & \text{якщо } t_{n_1}^1 \leq t_{n_2}^2 \\ t_{n_2}^2 & \text{якщо } t_{n_1}^1 > t_{n_2}^2 \end{cases} \quad (3.84)$$

$$d_n^1 = \begin{cases} a_{n_1}^1 & \text{якщо } t_{n_1}^1 \leq t_{n_2}^2 \\ \emptyset & \text{якщо } t_{n_1}^1 > t_{n_2}^2 \end{cases} \quad d_n^2 = \begin{cases} \emptyset & \text{якщо } t_{n_1}^1 < t_{n_2}^2 \\ a_{n_2}^2 & \text{якщо } t_{n_1}^1 \geq t_{n_2}^2 \end{cases}$$

Решта елементів визначається згідно з правилами БП1-БП59. Таким чином, у випадку відношення *відбувається між* при виконання синхронізації мультимодальних даних мультиобразів застосовується не окреме правило синхронізації, а шаблон синхронізації, що складається з правила синхронізації БП60 для визначення перших і останніх елементів та одного з правил БП1-БП59 для визначення всіх інших елементів.

Для зручності будемо використовувати наступний формат запису шаблону синхронізації (ШС), який визначимо за допомогою розширеної форми Бекуса-Наура (EBNF) [299]:

$$\begin{aligned} \text{шаблон\_синхронізації} = [ & \text{“ШС”}, \text{номер\_шаблону}, \text{“=”}, ] ( \text{“УП”} \mid \text{“БП”} \mid \text{“НП”} ), \\ & \text{номер\_правила}, [ \text{“ (“}, \text{номер\_елемента}, \text{ (“,”} \mid \text{“–”}, \\ & \text{номер\_елемента}, \text{”)”} \{ \text{“\&”}, ( \text{“УП”} \mid \text{“БП”} \mid \text{“НП”} ), \\ & \text{номер\_правила}, \text{“ (“}, \text{номер\_елемента}, \text{ (“,”} \mid \text{“–”}, \\ & \text{номер\_елемента}, \text{”)”} \} ] \end{aligned}$$

Розглянемо приклад запису шаблону синхронізації, визначеного згідно із запропонованим форматом: БП60(1, $n$ )&БП15(2–( $n$ -1)). Цей запис означає, що шаблон синхронізації складається з двох правил: правила БП60, яке застосовується для синхронізації перших та останніх елементів (їх індекси  $i = 1$  та  $i = n$  відповідно), та правила БП15, яке застосовується для синхронізації всіх інших елементів (ці елементи мають індекси  $i = [2 \dots (n - 1)]$ ).

Відповідно до визначеного формату шаблон синхронізації може мати власну назву, крім того, він може складатись з одного чи декількох правил синхронізації різних типів.

Таким чином, шаблон синхронізації для відношення *відбувається між* має вигляд: ШС1=БП60(1, $n$ )& $x$ (2–( $n$ -1)), де  $x$  – це деяке правило з базових правил БП1-БП59.

Визначимо правило синхронізації мультимодальних даних для випадку, коли отримання значень кортежу даних  $\bar{a}^1$  передуює отриманню значень кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *передуює*, що визначається (2.56). Отже, якщо  $b_d(\bar{t}^1, \bar{t}^2) = \text{true}$ , то  $\forall i_1 \in [1 \dots n_1], \forall i_2 \in [1 \dots n_2]$  та правило синхронізації мультимодальних даних БП61:

$$\begin{array}{ll}
t_{i_1} = t_{i_1}^1 & t_{i_2+n_1} = t_{i_2}^2 \\
d_{i_1}^1 = a_{i_1}^1 & d_{i_2+n_1}^1 = \emptyset \\
d_{i_1}^2 = \emptyset & d_{i_2+n_1}^2 = a_{i_2}^2
\end{array} \quad (3.85)$$

Випадок, коли отримання значень кортежу даних  $\bar{a}^1$  відбувається після отримання значень кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *настає після*, що визначається (2.57). Отже, якщо  $bi_d(\bar{t}^1, \bar{t}^2) = true$ , то  $\forall i_1 \in [1 \dots n_1], \forall i_2 \in [1 \dots n_2]$  та правило синхронізації мультимодальних даних БП62:

$$\begin{array}{ll}
t_{i_2} = t_{i_2}^2 & t_{i_1+n_2} = t_{i_1}^1 \\
d_{i_2}^1 = \emptyset & d_{i_1+n_2}^1 = a_{i_1}^1 \\
d_{i_2}^2 = a_{i_2}^2 & d_{i_1+n_2}^2 = \emptyset
\end{array} \quad (3.86)$$

Випадок, коли отримання значень кортежу даних  $\bar{a}^1$  передуює отриманню значень кортежу даних  $\bar{a}^2$ , проте момент отримання останнього значення кортежу даних  $\bar{a}^1$  збігається з моментом отримання першого значення кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *стикається з початком*, що визначається (2.59). Таким чином, якщо  $m_d(\bar{t}^1, \bar{t}^2) = true$ , то  $\forall i_1 \in [1 \dots (n_1 - 1)], \forall i_2 \in [2 \dots n_2]$  та правило синхронізації мультимодальних даних БП63:

$$\begin{array}{lll}
t_{i_1} = t_{i_1}^1 & t_{n_1} = t_{n_1}^1 & t_{i_2+n_1-1} = t_{i_2}^2 \\
d_{i_1}^1 = a_{i_1}^1 & d_{n_1}^1 = a_{n_1}^1 & d_{i_2+n_1-1}^1 = \emptyset \\
d_{i_1}^2 = \emptyset & d_{n_1}^2 = a_1^2 & d_{i_2+n_1-1}^2 = a_{i_2}^2
\end{array} \quad (3.87)$$

Випадок, коли отримання значень кортежу даних  $\bar{a}^1$  відбувається після отримання значень кортежу даних  $\bar{a}^2$ , проте момент отримання

першого значення кортежу даних  $\bar{a}^1$  збігається з моментом отримання останнього значення кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *стикається з кінцем*, що визначається (2.60). Отже, якщо  $mi_d(\bar{t}^1, \bar{t}^2) = true$ , то  $\forall i_1 \in [2 \dots n_1], \forall i_2 \in [1 \dots (n_2 - 1)]$  та правило синхронізації мультимодальних даних БП64:

$$\begin{array}{lll} t_{i_2} = t_{i_2}^2 & t_{n_2} = t_{n_2}^2 & t_{i_1+n_2-1} = t_{i_1}^1 \\ d_{i_2}^1 = \emptyset & d_{n_2}^1 = a_1^1 & d_{i_1+n_2-1}^1 = a_{i_1}^1 \\ d_{i_2}^2 = a_{i_2}^2 & d_{n_2}^2 = a_{n_2}^2 & d_{i_1+n_2-1}^2 = \emptyset \end{array} \quad (3.88)$$

Випадок, коли отримання значень кортежу даних  $\bar{a}^1$  передуює отриманню значень кортежу даних  $\bar{a}^2$ , проте отримання  $K$  останніх значень кортежу даних  $\bar{a}^1$  збігається з отриманням  $K$  перших значень кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *перекриває*, що визначається (2.61). Тоді, якщо  $o_d(\bar{t}^1, \bar{t}^2) = true$ , то  $\forall i_1 \in [1 \dots (n_1 - K)], \forall i_2 \in [(K + 1) \dots n_2], \forall k \in [1 \dots K]$  та правило синхронізації мультимодальних даних БП65:

$$\begin{array}{lll} t_{i_1} = t_{i_1}^1 & t_{n_1+k-K} = t_k^2 & t_{n_1+i_2-K} = t_{i_2}^2 \\ d_{i_1}^1 = a_{i_1}^1 & d_{n_1+k-K}^1 = a_{n_1+k-K}^1 & d_{n_1+i_2-K}^1 = \emptyset \\ d_{i_1}^2 = \emptyset & d_{n_1+k-K}^2 = a_k^2 & d_{n_1+i_2-K}^2 = a_{i_2}^2 \end{array} \quad (3.89)$$

Випадок, коли отримання значень кортежу даних  $\bar{a}^1$  відбувається після отримання значень кортежу даних  $\bar{a}^2$ , проте отримання  $K$  перших значень кортежу даних  $\bar{a}^1$  збігається з отриманням  $K$  останніх значень кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *перекривається*, що визначається (2.62). Таким чином, якщо

$oi_d(\bar{t}^1, \bar{t}^2) = true$ , то  $\forall i_1 \in [(K+1) \dots n_1], \forall i_2 \in [1 \dots (n_2 - K)], \forall k \in [1 \dots K]$  та правило синхронізації мультимодальних даних БП66:

$$\begin{array}{lll} t_{i_2} = t_{i_2}^2 & t_{n_2+k-K} = t_k^1 & t_{i_1+n_2-K} = t_{i_1}^1 \\ d_{i_2}^1 = \emptyset & d_{n_2+k-K}^1 = a_k^1 & d_{i_1+n_2-K}^1 = a_{i_1}^1 \\ d_{i_2}^2 = a_{i_2}^2 & d_{n_2+k-K}^2 = a_{n_2+k-K}^2 & d_{i_1+n_2-K}^2 = \emptyset \end{array} \quad (3.90)$$

Якщо отримання значень кортежу даних  $\bar{a}^1$  починається пізніше та завершується раніше отримання значень кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *відбувається під час*, що визначається (2.63). Отже, якщо  $d_d(\bar{t}^1, \bar{t}^2) = true$ , то  $\bar{t}^2 = \bar{t}^{2,I} \cup \bar{t}^{2,II} \cup \bar{t}^{2,III}$ , такі, що  $b_d(\bar{t}^{2,I}, \bar{t}^1), w_d^{(t_1^1, t_{n_1}^1)}(\bar{t}^{2,II}, \bar{t}^1), bi_d(\bar{t}^{2,III}, \bar{t}^1)$ . Тоді,  $\forall k_1 \in [1 \dots K_1], \forall k_2 \in [K_2 \dots n_2]$ , таких, що  $t_{K_1}^2 < t_1^1$  та  $t_{K_2}^2 > t_{n_1}^1$ , правило синхронізації БП67 для отримання перших значень  $t_{k_1} (t_{k_1} \in \bar{t}^{2,I}), d_{k_1}^1, d_{k_1}^2$  та останніх значень  $t_{k_2} (t_{k_2} \in \bar{t}^{2,III}), d_{k_2}^1, d_{k_2}^2$  визначається виразами (3.91). Решту елементів  $(t_k \in \bar{t}^{2,II}, d_k^1, d_k^2)$  кортежів  $\bar{t}, \bar{d}^1$  та  $\bar{d}^2$  можна отримати згідно з правилами синхронізації БП1-БП59, тобто синхронізація даних мультимедіа у випадку відношення *відбувається під час* задаються шаблоном синхронізації ШС2=БП67(1,n)&x(2-(n-1)), де  $x$  – це деяке правило з базових правил БП1-БП59. Довжина кортежу  $\bar{t}$  складає  $n = K_1 + m + n_2 - K_2 + 1$ , де  $m$  є довжиною кортежу, отриманого в результаті синхронізації кортежів  $\bar{t}^{2,II}$  та  $\bar{t}^1$ .

$$\begin{array}{ll} t_{k_1} = t_{k_1}^2 & t_{n-n_2+k_2} = t_{k_2}^2 \\ d_{k_1}^1 = \emptyset & d_{n-n_2+k_2}^1 = \emptyset \\ d_{k_1}^2 = a_{k_1}^2 & d_{n-n_2+k_2}^2 = a_{k_2}^2 \end{array} \quad (3.91)$$

Випадок, коли отримання значень кортежу даних  $\bar{a}^1$  починається раніше та завершується пізніше отримання значень кортежу даних  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *містить*, що визначається (2.64). Отже, якщо  $di_d(\bar{t}^1, \bar{t}^2)$ , то  $\bar{t}^1 = \bar{t}^{1,I} \cup \bar{t}^{1,II} \cup \bar{t}^{1,III}$ , такі, що  $b_d(\bar{t}^{1,I}, \bar{t}^2)$ ,  $w_d^{(t_1^2, t_{n_2}^2)}(\bar{t}^{1,II}, \bar{t}^2)$ ,  $bi_d(\bar{t}^{1,III}, \bar{t}^2)$ . Тоді,  $\forall k_1 \in [1 \dots K_1]$ ,  $\forall k_2 \in [K_2 \dots n_1]$ , таких, що  $t_{K_1}^1 < t_1^2$  та  $t_{K_2}^1 > t_{n_2}^2$ , правило синхронізації БП68 для отримання перших значень  $t_{k_1}$  ( $t_{k_1} \in \bar{t}^{1,I}$ ),  $d_{k_1}^1$ ,  $d_{k_1}^2$  та останніх значень  $t_{k_2}$  ( $t_{k_2} \in \bar{t}^{1,III}$ ),  $d_{k_2}^1$ ,  $d_{k_2}^2$  визначається виразами (3.92). Решту елементів ( $t_k \in \bar{t}^{1,II}$ ,  $d_k^1$ ,  $d_k^2$ ) кортежів  $\bar{t}$ ,  $\bar{d}^1$  та  $\bar{d}^2$  можна отримати згідно з правилами синхронізації БП1-БП59, тобто синхронізація даних мультимедіа у разі відношення *містить* задаються шаблоном синхронізації ШСЗ=БП68(1,n)&x(2-(n-1)), де x – це деяке правило з правил БП1-БП59. Довжина кортежу  $\bar{t}$  складає  $n = K_1 + m + n_1 - K_2 + 1$ , де m є довжиною кортежу, отриманого в результаті синхронізації  $\bar{t}^{1,II}$  та  $\bar{t}^2$ .

$$\begin{aligned} t_{k_1} &= t_{k_1}^1 & t_{n-n_1+k_2} &= t_{k_2}^1 \\ d_{k_1}^1 &= a_{k_1}^1 & d_{n-n_1+k_2}^1 &= a_{k_2}^1 \\ d_{k_1}^2 &= \emptyset & d_{n-n_1+k_2}^2 &= \emptyset \end{aligned} \quad (3.92)$$

Якщо отримання значень кортежів даних  $\bar{a}^1$  та  $\bar{a}^2$  починається одночасно, проте отримання значень кортежу  $\bar{a}^1$  закінчується раніше, ніж отримання значень кортежу  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *починає*, що визначається (2.65). Отже, якщо  $sd_d(\bar{t}^1, \bar{t}^2)$ , то  $\bar{t}^2 = \bar{t}^{2,I} \cup \bar{t}^{2,II}$ , такі, що  $w_d^{(t_1^1, t_{n_1}^1)}(\bar{t}^{2,I}, \bar{t}^1)$ ,  $bi_d(\bar{t}^{2,II}, \bar{t}^1)$ . Тоді,  $\forall k_2 \in [K \dots n_2]$ , таких, що  $t_K^2 > t_{n_1}^1$ , правило синхронізації БП69 для отримання першого значення  $t_1$  ( $t_1 \in \bar{t}^{2,I}$ ),  $d_1^1$ ,  $d_1^2$  та останніх значень  $t_{k_2}$  ( $t_{k_2} \in \bar{t}^{2,II}$ ),  $d_{k_2}^1$ ,  $d_{k_2}^2$  результуючих кортежів  $\bar{t}$ ,  $\bar{d}^1$  та  $\bar{d}^2$  визначається



виразами (3.93). Решту елементів ( $t_{k_1} \in \overline{t^{2,I}}, d_{k_1}^1, d_{k_1}^2, \forall k_1 \in [2 \dots m]$ , де  $m$  є довжиною кортежу, отриманого в результаті синхронізації кортежів  $\overline{t^{2,I}}$  та  $\overline{t^1}$ ) кортежів  $\bar{t}$ ,  $\bar{d}^1$  та  $\bar{d}^2$  можна отримати згідно з правилами синхронізації БП1-БП59, тобто синхронізація мультимодальних даних мультиобразів у випадку відношення *починає* задаються шаблоном синхронізації ШС4=БП69(1,n)&x(2-(n-1)), де  $x$  – це деяке правило з базових правил БП1-БП59. Довжина кортежу  $\bar{t}$  складає  $n = m + n_2 - K + 1$ .

$$\begin{array}{ll} t_1 = t_1^1 & t_{n-n_2+k_2} = t_{k_2}^2 \\ d_1^1 = a_1^1 & d_{n-n_2+k_2}^1 = \emptyset \\ d_1^2 = a_1^2 & d_{n-n_2+k_2}^2 = a_{k_2}^2 \end{array} \quad (3.93)$$

Випадок, коли отримання значень кортежів даних  $\bar{a}^1$  та  $\bar{a}^2$  починається одночасно, проте отримання значень кортежу  $\bar{a}^1$  закінчується пізніше, ніж отримання значень кортежу  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *починається*, що визначається (2.66). Отже, якщо  $si_d(\bar{t}^1, \bar{t}^2)$ , то  $\bar{t}^1 = \overline{t^{1,I}} \cup \overline{t^{1,II}}$ , такі, що  $w_d^{(t_1^2, t_{n_2}^2)}(\overline{t^{1,I}}, \bar{t}^2)$ ,  $bi_d(\overline{t^{1,II}}, \bar{t}^2)$ . Тоді,  $\forall k_2 \in [K \dots n_1]$ , таких, що  $t_K^1 > t_{n_2}^2$ , правило синхронізації БП70 для отримання першого значення  $t_1$  ( $t_1 \in \overline{t^{1,I}}$ ),  $d_1^1, d_1^2$  та останніх значень  $t_{k_2}$  ( $t_{k_2} \in \overline{t^{1,II}}$ ),  $d_{k_2}^1, d_{k_2}^2$  результуючих кортежів  $\bar{t}$ ,  $\bar{d}^1$  та  $\bar{d}^2$  визначається виразами (3.94). Решту елементів ( $t_{k_1} \in \overline{t^{1,I}}, d_{k_1}^1, d_{k_1}^2, \forall k_1 \in [2 \dots m]$ , де  $m$  є довжиною кортежу, отриманого в результаті синхронізації кортежів  $\overline{t^{1,I}}$  та  $\bar{t}^2$ ) кортежів  $\bar{t}$ ,  $\bar{d}^1$  та  $\bar{d}^2$  можна отримати згідно з правилами синхронізації БП1-БП59, тобто синхронізація мультимодальних даних мультиобразів у випадку відношення *починається* задається шаблоном синхронізації ШС5=БП70(1,n)&x(2-(n-1)), де  $x$  – це деяке правило з базових правил БП1-БП59. Довжина кортежу  $\bar{t}$  складає  $n = m + n_1 - K + 1$ .

$$\begin{array}{ll}
t_1 = t_1^1 & t_{n-n_1+k_2} = t_{k_2}^1 \\
d_1^1 = a_1^1 & d_{n-n_1+k_2}^1 = a_{k_2}^1 \\
d_1^2 = a_1^2 & d_{n-n_1+k_2}^2 = \emptyset
\end{array} \quad (3.94)$$

Випадок, коли отримання значень кортежів даних  $\bar{a}^1$  та  $\bar{a}^2$  закінчується одночасно, проте отримання значень кортежу  $\bar{a}^1$  починається пізніше, ніж отримання значень кортежу  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *закінчує*, що визначається (2.67). Отже, якщо  $f_d(\bar{t}^1, \bar{t}^2)$ , то  $\bar{t}^2 = \bar{t}^{2,I} \cup \bar{t}^{2,II}$ , такі, що  $b_d(\bar{t}^{2,I}, \bar{t}^1), w_d^{(t_1^1, t_{n_1}^1)}(\bar{t}^{2,II}, \bar{t}^1)$ . Тоді,  $\forall k_1 \in [1 \dots K]$ , таких, що  $t_{k_1}^2 < t_1^1$ , правило синхронізації БП71 для отримання перших значень  $t_{k_1} (t_{k_1} \in \bar{t}^{2,I})$ ,  $d_{k_1}^1, d_{k_1}^2$  та останнього значення  $t_n (t_n \in \bar{t}^{2,II})$ ,  $d_n^1, d_n^2$  результуючих кортежів  $\bar{t}, \bar{d}^1$  та  $\bar{d}^2$  визначається виразами (3.95). Решту елементів ( $t_{k_2} \in \bar{t}^{2,II}, d_{k_2}^1, d_{k_2}^2, \forall k_2 \in [(K+1) \dots (n-1)]$ ), де довжина кортежу  $\bar{t}$  складає  $n = K + m$ ;  $m$  є довжиною кортежу, отриманого в результаті синхронізації кортежів  $\bar{t}^{2,II}$  та  $\bar{t}^1$ ) кортежів  $\bar{t}, \bar{d}^1$  та  $\bar{d}^2$  можна отримати згідно з правилами синхронізації БП1-БП59, тобто синхронізація даних мультимедіа у разі відношення *закінчує* задаються шаблоном ШС6=БП71(1,n)&x(2-(n-1)), де  $x$  – це деяке правило з базових правил БП1-БП59.

$$\begin{array}{ll}
t_{k_1} = t_{k_1}^2 & t_n = t_{n_1}^1 \\
d_{k_1}^1 = \emptyset & d_n^1 = a_{n_1}^1 \\
d_{k_1}^2 = a_{k_1}^2 & d_n^2 = a_{n_2}^2
\end{array} \quad (3.95)$$

Випадок, коли отримання значень кортежів даних  $\bar{a}^1$  та  $\bar{a}^2$  закінчується одночасно, проте отримання значень кортежу  $\bar{a}^1$  починається

раніше, ніж отримання значень кортежу  $\bar{a}^2$ , означає, що дискретні інтервали  $\bar{t}^1$  та  $\bar{t}^2$  пов'язані відношенням *закінчується*, що визначається (2.68). Отже, якщо  $fi_d(\bar{t}^1, \bar{t}^2)$ , то  $\bar{t}^1 = \bar{t}^{1,I} \cup \bar{t}^{1,II}$ , такі, що  $b_d(\bar{t}^{1,I}, \bar{t}^2)$ ,  $w_d^{(t_1^2, t_{n_2}^2)}(\bar{t}^{1,II}, \bar{t}^2)$ . Тоді,  $\forall k_1 \in [1 \dots K]$ , таких, що  $t_K^1 < t_1^2$ , правило синхронізації БП72 для отримання перших значень  $t_{k_1}$  ( $t_{k_1} \in \bar{t}^{1,I}$ ),  $d_{k_1}^1$ ,  $d_{k_1}^2$  та останнього значення  $t_n$  ( $t_n \in \bar{t}^{1,II}$ ),  $d_n^1$ ,  $d_n^2$  результуючих кортежів  $\bar{t}$ ,  $\bar{d}^1$  та  $\bar{d}^2$  визначається виразами (3.96). Решту елементів ( $t_{k_2} \in \bar{t}^{1,II}$ ,  $d_{k_2}^1$ ,  $d_{k_2}^2$ ),  $\forall k_2 \in [(K + 1) \dots (n - 1)]$ , де довжина кортежу  $\bar{t}$  складає  $n = K + m$ ;  $m$  є довжиною кортежу, отриманого в результаті синхронізації кортежів  $\bar{t}^{1,II}$  та  $\bar{t}^2$ ) кортежів  $\bar{t}$ ,  $\bar{d}^1$  та  $\bar{d}^2$  можна отримати згідно з правилами синхронізації БП1-БП59, тобто синхронізація даних мультиобразів у випадку відношення *закінчується* задаються шаблоном синхронізації ШС7=БП72(1,n)&x(2-(n-1)), де  $x$  – це деяке правило з базових правил БП1-БП59.

$$\begin{array}{ll}
 t_{k_1} = t_{k_1}^1 & t_n = t_{n_2}^2 \\
 d_{k_1}^1 = a_{k_1}^1 & d_n^1 = a_{n_1}^1 \\
 d_{k_1}^2 = \emptyset & d_n^2 = a_{n_2}^2
 \end{array} \quad (3.96)$$

Оскільки існує значна кількість випадків точної синхронізації даних, то для спрощення процедури синхронізації мультимодальних даних у задачах, де визначення часу з деякою похибкою є припустимим, доцільно застосовувати нечітку синхронізацію даних.

### 3.3.2. Правила нечіткої синхронізації

Для нечіткої синхронізації даних потрібно визначити нечіткий еквівалент інтервальних відношень, визначених (2.56)-(2.69). Формальному

визначенню нечітких відношень присвячено багато робіт, зокрема, [270, 272–276]. Застосуємо підхід, сформульований у роботах [270] та [274].

Ступінь  $bb_{(\alpha,\beta)}^{\leq}(\bar{t}^1, \bar{t}^2)$ , до якого початок кортежу  $\bar{t}^1$  приблизно на  $\alpha$  ( $\alpha \in \mathbb{R}$ ) часових відліків випереджає початок кортежу  $\bar{t}^2$ , де  $\beta$  ( $\beta \geq 0$ ) визначає припустиму похибку, є узагальненням нечіткого подання (*fuzzification*) строгої нерівності  $t_1^1 < t_1^2$  та може бути визначене як:

$$bb_{(\alpha,\beta)}^{\leq}(\bar{t}^1, \bar{t}^2) = \bar{t}^1 \circ_{T_w} (L_{(\alpha,\beta)}^{\leq} \triangleright_{I_{T_w}} \bar{t}^2), \quad (3.97)$$

де операція  $\circ_T$  означає композицію нечітких відношень, визначених (3.98); оператор  $\triangleright_I$  означає *superproduct*-композицію, визначену (3.99) згідно з [277];  $T_w$  – t-норма Лукашевича, визначена (3.100);  $I_{T_w}$  – імплікатор Лукашевича, визначений (3.101);  $L_{(\alpha,\beta)}^{\leq}$  – нечітке відношення між моментами часу  $t_1^1$  та  $t_1^2$ , визначене (3.102).

Sup- $T$  композиція нечітких відношень визначається як:

$$(R_1 \circ_T R_2)(x, z) = \sup_{y \in Y} T(R_1(x, y), R_2(y, z)), \quad (3.98)$$

де  $R_1$  – нечітке відношення з множини  $X$  до множини  $Y$ ;  $R_2$  – нечітке відношення з множини  $Y$  до множини  $Z$ ;  $x \in X, y \in Y, z \in Z$ .

$I$ -*superproduct*-композиція нечітких відношень визначається як:

$$R_1 \triangleright_I R_2(x, z) = \inf_{y \in Y} I(R_2(y, z), R_1(x, y)) \quad (3.99)$$

$T$ -норма Лукашевича та її імплікатор визначаються як:

$$T_w(a, b) = \max(0, a + b - 1), \quad (3.100)$$

де  $a, b \in [0, 1]$ .

$$I_{T_w}(a, b) = \min(1 - a + b, 1), \quad (3.101)$$

Нечітке відношення  $L_{(\alpha, \beta)}^{\ll}$  моделює відмінність між моментами часу та визначається як:

$$L_{(\alpha, \beta)}^{\ll}(t_1^1, t_1^2) = \begin{cases} 1 & \text{якщо } t_1^2 - t_1^1 > \alpha + \beta \\ 0 & \text{якщо } t_1^2 - t_1^1 \leq \alpha \\ \frac{t_1^2 - t_1^1 - \alpha}{\beta} & \text{в інших випадках} \end{cases}, \quad (3.102)$$

де  $\alpha$  – різниця (в одиницях часу) між моментами часу  $t_1^1$  та  $t_1^2$ ;  $\beta$  – припустима похибка.

Ступінь  $bb_{(\alpha, \beta)}^{\leq}(\bar{t}^1, \bar{t}^2)$ , до якого початок кортежу  $\bar{t}^1$  приблизно на  $\alpha$  часових відліків випереджає або збігається з початком кортежу  $\bar{t}^2$ , є узагальненням нечіткого подання нестрогої нерівності  $t_1^1 \leq t_1^2$  та може бути визначене як:

$$bb_{(\alpha, \beta)}^{\leq}(\bar{t}^1, \bar{t}^2) = 1 - bb_{(\alpha, \beta)}^{\ll}(\bar{t}^2, \bar{t}^1) \quad (3.103)$$

Ступінь  $ee_{(\alpha, \beta)}^{\ll}(\bar{t}^1, \bar{t}^2)$ , до якого кінець кортежу  $\bar{t}^1$  приблизно на  $\alpha$  часових відліків випереджає кінець кортежу  $\bar{t}^2$ , є узагальненням нечіткого подання строгої нерівності  $t_{n_1}^1 < t_{n_2}^2$  та може бути визначене як:

$$ee_{(\alpha, \beta)}^{\ll}(\bar{t}^1, \bar{t}^2) = (\bar{t}^1 \triangleleft_{I_{T_w}} L_{(\alpha, \beta)}^{\ll}) \circ_{T_w} \bar{t}^2, \quad (3.104)$$

де оператор  $\triangleleft_I$  означає *subproduct*-композицію, визначену (3.105) відповідно до [277].

$$R_1 \triangleleft_I R_2(x, z) = \inf_{y \in Y} I(R_1(x, y), R_2(y, z)) \quad (3.105)$$

Ступінь  $ee_{(\alpha, \beta)}^{\leq}(\bar{t}^1, \bar{t}^2)$ , до якого кінець кортежу  $\bar{t}^1$  приблизно на  $\alpha$  часових відліків випереджає або збігається з кінцем кортежу  $\bar{t}^2$  є узагальненням нечіткого подання нестрогої нерівності  $t_{n_1}^1 \leq t_{n_2}^2$  та може бути визначене як:

$$ee_{(\alpha, \beta)}^{\leq}(\bar{t}^1, \bar{t}^2) = 1 - ee_{(\alpha, \beta)}^{\ll}(\bar{t}^2, \bar{t}^1) \quad (3.106)$$

Ступінь  $be_{(\alpha, \beta)}^{\ll}(\bar{t}^1, \bar{t}^2)$ , до якого початок кортежу  $\bar{t}^1$  приблизно на  $\alpha$  часових відліків випереджає кінець кортежу  $\bar{t}^2$  є узагальненням нечіткого подання строгої нерівності  $t_1^1 < t_{n_2}^2$  та може бути визначене як:

$$be_{(\alpha, \beta)}^{\ll}(\bar{t}^1, \bar{t}^2) = \bar{t}^1 \circ_{T_w} L_{(\alpha, \beta)}^{\ll} \circ_{T_w} \bar{t}^2 \quad (3.107)$$

Ступінь  $be_{(\alpha, \beta)}^{\leq}(\bar{t}^1, \bar{t}^2)$ , до якого початок кортежу  $\bar{t}^1$  приблизно на  $\alpha$  часових відліків випереджає або збігається з кінцем кортежу  $\bar{t}^2$  є узагальненням нечіткого подання нестрогої нерівності  $t_1^1 \leq t_{n_2}^2$  та може бути визначене як:

$$be_{(\alpha, \beta)}^{\leq}(\bar{t}^1, \bar{t}^2) = 1 - be_{(\alpha, \beta)}^{\ll}(\bar{t}^2, \bar{t}^1) \quad (3.108)$$

Ступінь  $eb_{(\alpha, \beta)}^{\ll}(\bar{t}^1, \bar{t}^2)$ , до якого кінець кортежу  $\bar{t}^1$  приблизно на  $\alpha$  часових відліків випереджає початок кортежу  $\bar{t}^2$ , є узагальненням нечіткого подання строгої нерівності  $t_{n_1}^1 < t_1^2$  та може бути визначене як:

$$eb_{(\alpha,\beta)}^{\leq}(\bar{t}^1, \bar{t}^2) = \bar{t}^1 \triangleleft_{I_{T_w}} L_{(\alpha,\beta)}^{\leq} \triangleright_{I_{T_w}} \bar{t}^2 \quad (3.109)$$

Ступінь  $eb_{(\alpha,\beta)}^{\leq}(\bar{t}^1, \bar{t}^2)$ , до якого кінець кортежу  $\bar{t}^1$  приблизно на  $\alpha$  часових відліків випереджає або збігається з початком кортежу  $\bar{t}^2$  є узагальненням нечіткого подання нестрогої нерівності  $t_{n_1}^1 \leq t_1^2$  та може бути визначене як:

$$eb_{(\alpha,\beta)}^{\leq}(\bar{t}^1, \bar{t}^2) = 1 - eb_{(\alpha,\beta)}^{\leq}(\bar{t}^2, \bar{t}^1) \quad (3.110)$$

Будь-який з розглянутих ступенів (3.97), (3.103), (3.104), (3.106)-(3.110), зокрема,  $eb_{(\alpha,\beta)}^{\leq}(t_i^1, t_i^2)$ , може бути застосований для визначення близькості значень моментів часу  $t_i^1$  та  $t_i^2$ , що дозволить спростити синхронізацію даних.

### Теорема 3.2 (теорема про похибку).

Нехай  $\Delta t$  – це похибка визначення часу отримання даних, що очікуються у моменти часу  $t_i^1$  та  $t_i^2$ , тобто  $\Delta t$  є мінімальною відстанню між  $t_i^1$  та  $t_i^2$ , тоді  $eb_{(\alpha,\beta)}^{\leq}(t_i^1, t_i^2) = 0$  та моменти часу  $t_i^1$  та  $t_i^2$  є єдиним моментом нечіткого часу  $\tilde{t}_i$ .

Доведення:

$$\text{Згідно з (3.109), } eb_{(\alpha,\beta)}^{\leq}(t_i^1, t_i^2) = t_i^1 \triangleleft_{I_{T_w}} L_{(\alpha,\beta)}^{\leq}(t_i^1, t_i^2) \triangleright_{I_{T_w}} t_i^2.$$

З виразів (3.99) та (3.105), отримуємо:

$$eb_{(\alpha,\beta)}^{\leq}(t_i^1, t_i^2) = \inf_{t_i^1 \in \bar{t}^1} \left( \inf_{t_i^2 \in \bar{t}^2} I_{T_w} \left( t_i^2, I_{T_w} \left( t_i^1, L_{(\alpha,\beta)}^{\leq}(t_i^1, t_i^2) \right) \right) \right) \quad (3.111)$$

Враховуючи (3.101), отримуємо:

$$I_{T_w}(t_i^1, L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2)) = \min(1 - 1 + L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2), 1) = L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) \quad (3.112)$$

Аналогічно,

$$I_{T_w}(t_i^2, L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2)) = \min(1 - 1 + L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2), 1) = L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) \quad (3.113)$$

Тоді, можемо представити (3.111) у вигляді:

$$eb_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) = \inf_{t_i^1 \in \bar{t}^1} \left( \inf_{t_i^2 \in \bar{t}^2} L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) \right) \quad (3.114)$$

Згідно з (3.102):

$$L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) = \begin{cases} 1 & \text{якщо } t_i^2 - t_i^1 > \alpha + \beta \\ 0 & \text{якщо } t_i^2 - t_i^1 \leq \alpha \\ \frac{t_i^2 - t_i^1 - \alpha}{\beta} & \text{в інших випадках} \end{cases} \quad (3.115)$$

Оскільки  $t_i^2 = t_i^1 + \Delta t$  та  $\Delta t$  є мінімальною відстанню між  $t_i^1$  та  $t_i^2$ , то в (3.97)  $t_i^2 - t_i^1 < \alpha$ .

Тоді,  $L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) = 0$  та, внаслідок цього,  $eb_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) = 0$ , тобто моменти часу  $t_i^1$  та  $t_i^2$  є єдиним моментом нечіткого часу  $\tilde{t}_i$ . Що і потрібно було довести. ■

Наслідок з теореми 3.2. Нехай відстань між моментами часу  $t_i^1$  та  $t_i^2$  перевищує  $\Delta t$  та дорівнює  $\alpha$ . Тоді, в (3.115) отримуємо  $t_i^2 - t_i^1 = \alpha$ , тому,  $L_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) = 0$  та, внаслідок цього,  $eb_{(\alpha, \beta)}^{\ll}(t_i^1, t_i^2) = 0$ . Таким чином,



моменти часу  $t_i^1$  та  $t_i^2$  можуть вважатись єдиним моментом нечіткого часу  $\tilde{t}_i$ .

Практичне значення теореми про похибку полягає у тому, що, якщо похибка вимірювання часу у двох послідовностях часових значень  $\bar{t}^1$  та  $\bar{t}^2$  не перевищує мінімальної різниці часу, що може бути зафіксована у програмно-апаратній системі, то цією похибкою можна знехтувати та замінити  $\bar{t}^1$  та  $\bar{t}^2$  єдиним значенням. Це дозволяє спростити синхронізацію даних.

Розглянемо можливі випадки (рис. 3.6) для значення  $\alpha$  між сусідніми значеннями в  $\bar{t}^1$  та  $\bar{t}^2$ , коли  $eb_{(\alpha,\beta)}^{\leq}(t_i^1, t_i^2) = 0$  та визначимо шляхи спрощення правил синхронізації даних БП1-БП59.

Нехай значення  $\alpha_1$  є відстанню між  $t_i^1$  та  $t_i^2$ , значення  $\alpha_2$  є відстанню між  $t_i^2$  та  $t_{i+1}^1$ , та значення  $\alpha_3$  є відстанню між  $t_{i+1}^1$  та  $t_{i+1}^2$ . Крім цього, нехай значення  $\alpha$  є мінімальною відстанню між такими  $t_i^1$  та  $t_i^2$ , які можуть вважатись одним значенням нечіткого часу  $\tilde{t}$ .

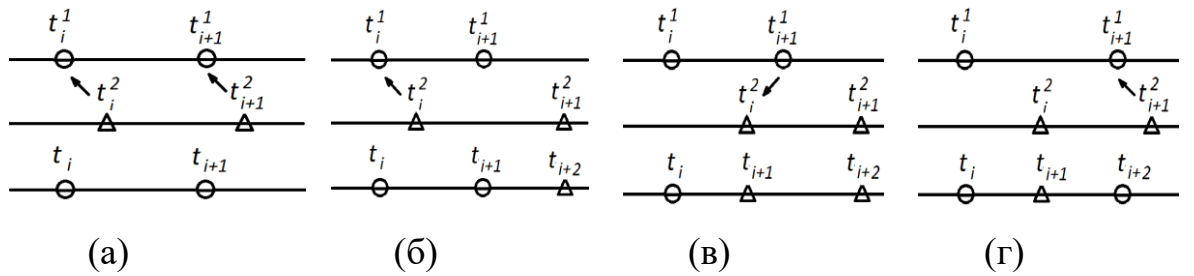


Рис. 3.6. Випадки нечіткої синхронізації

Якщо обидві нерівності  $\alpha_1 < \alpha$  та  $\alpha_3 < \alpha$  виконуються (рис. 3.6а), то незалежно від значення  $\alpha_2$  правило нечіткої синхронізації даних НП1 може бути сформульоване у вигляді:

$$\begin{array}{ll}
\tilde{t}_i = t_i^1 & \tilde{t}_{i+1} = t_{i+1}^1 \\
d_i^1 = a_i^1 & d_{i+1}^1 = a_{i+1}^1 \\
d_i^2 = a_i^2 & d_{i+1}^2 = a_{i+1}^2
\end{array} \quad (3.116)$$

Якщо обидві нерівності  $\alpha_1 < \alpha$  та  $\alpha_3 > \alpha$  виконуються (рис. 3.6б), то незалежно від значення  $\alpha_2$  правило нечіткої синхронізації даних НП2 може бути сформульоване у вигляді:

$$\begin{array}{lll}
\tilde{t}_i = t_i^1 & t_{i+1} = t_{i+1}^1 & t_{i+2} = t_{i+1}^2 \\
d_i^1 = a_i^1 & d_{i+1}^1 = a_{i+1}^1 & d_{i+2}^1 = \emptyset \\
d_i^2 = a_i^2 & d_{i+1}^2 = \emptyset & d_{i+2}^2 = a_{i+1}^2
\end{array} \quad (3.117)$$

Якщо обидві нерівності  $\alpha_1 > \alpha$  та  $\alpha_2 < \alpha$  виконуються (рис. 3.6в), то незалежно від значення  $\alpha_3$  правило нечіткої синхронізації даних НП3 може бути сформульоване у вигляді:

$$\begin{array}{lll}
t_i = t_i^1 & \tilde{t}_{i+1} = t_i^2 & t_{i+2} = t_{i+1}^2 \\
d_i^1 = a_i^1 & d_{i+1}^1 = a_{i+1}^1 & d_{i+2}^1 = \emptyset \\
d_i^2 = \emptyset & d_{i+1}^2 = a_i^2 & d_{i+2}^2 = a_{i+1}^2
\end{array} \quad (3.118)$$

Якщо нерівність  $\alpha_1 > \alpha > \alpha_2 > \alpha_3$  або обидві нерівності  $\alpha_1 > \alpha > \alpha_3$  та  $\alpha_2 > \alpha$  виконуються (рис. 3.6г), то правило нечіткої синхронізації даних НП4 може бути сформульоване у вигляді:

$$\begin{array}{lll}
t_i = t_i^1 & t_{i+1} = t_i^2 & \tilde{t}_{i+2} = t_{i+1}^1 \\
d_i^1 = a_i^1 & d_{i+1}^1 = \emptyset & d_{i+2}^1 = a_{i+1}^1 \\
d_i^2 = \emptyset & d_{i+1}^2 = a_i^2 & d_{i+2}^2 = a_{i+1}^2
\end{array} \quad (3.119)$$

Таким чином, нечітка синхронізація дозволяє спростити узгодження часових значень та відповідних даних в разі, якщо ці дані отримані у

настільки близькі моменти часу, що можна знехтувати відмінністю між ними. Ступінь близькості моментів часу  $t_i^1$  та  $t_i^2$  задається умовами вирішуваної задачі та може бути визначена не лише ступенем  $eb_{(\alpha,\beta)}^{\ll}(t_i^1, t_i^2)$ , але й іншими ступенями відношення близькості, визначеними (3.97), (3.103), (3.104), (3.106)-(3.108), (3.110).

### 3.3.3. Процедура синхронізації мультиобразів

Ґрунтуючись на інтервальних відношеннях (2.91)-(2.92) та визначених правилах синхронізації (УП1, БП1-БП72, НП1-НП4) та шаблонах синхронізації (ШС1-ШС7), сформулюємо метод синхронізації мультиобразів, зокрема, процедуру синхронізації, що лежить у його основі.

Вхідними даними для метода є мультиобрази  $I_1, \dots, I_K$  ( $K \geq 2$ ), припустима похибка визначення часу  $\Delta t$  та тип кожної послідовності часових значень (періодична / неперіодична). Результатом методу є синхронізовані мультиобрази  $I_1^S, \dots, I_K^S$ . Під *синхронізованими* будемо розуміти мультиобрази, для яких виконуються такі умови:

- всі кортежі всіх мультиобразів мають однакову довжину;
- часові кортежі у всіх мультиобразах є однаковими (рівними);
- всі значення у кортежах мультимодальних даних є узгодженими зі значеннями часових кортежів так, що  $i$ -тий елемент у кожному кортежі кожного мультиобразу відповідає  $i$ -тому елементу у часовому кортежі будь-якого мультиобразу.

Метод передбачає виконання декількох проходів (повторень всіх етапів). Кількість проходів  $P$  залежить від кількості  $K$  мультиобразів, які потрібно синхронізувати:

$$P = \lceil \log_2 K \rceil, \quad (3.120)$$

де  $[x]$  – операція округлення  $x$  до найближчого цілого, більшого за  $x$ .

На кожному з  $P$  проходів виконуються два етапи.

Перший етап розпочинається з визначення довжини  $n_k$  часового кортежу  $\langle t_i^{(k)} \rangle_{i=1}^{n_k}$  кожного мультиобразу  $I_k$ :

$$I_k = \left[ \left[ T, M_1^{(k)}, \dots, M_{N_k}^{(k)} \mid \langle t_i^{(k)} \rangle_{i=1}^{n_k}, \langle a_i^{(k),1} \rangle_{i=1}^{n_k}, \dots, \langle a_i^{(k),N_k} \rangle_{i=1}^{n_k} \right] \right], \quad (3.121)$$

де  $k = [1 \dots K]$ ;  $K$  – кількість мультиобразів, що потребують синхронізації.

Далі список із  $K$  мультиобразів впорядковується за зростанням довжини часового кортежу та поділяється на дві частини, що утворюють дві групи мультиобразів: до першої групи входять  $\left\lfloor \frac{K}{2} \right\rfloor$  мультиобразів з меншою довжиною часового кортежу, до другої групи входять  $\left\lceil \frac{K}{2} \right\rceil$  мультиобразів з більшою довжиною часового кортежу, де  $[x]$  – операція округлення  $x$  до найближчого цілого, меншого за  $x$ ,  $\lceil x \rceil$  – операція округлення  $x$  до найближчого цілого, більшого за  $x$ .

На другому етапі виконується процедура попарної синхронізації мультиобразів. Пара мультиобразів утворюється з  $p$ -го мультиобразу першої групи  $I_{p,I}$  та  $p$ -го мультиобразу другої групи  $I_{p,II}$ , де  $p = \left[ 1 \dots \left\lceil \frac{K}{2} \right\rceil \right]$ , тобто маємо пару  $\langle I_{p,I}, I_{p,II} \rangle$ . При цьому, якщо  $K \bmod 2 \neq 0$ , то непарний мультиобраз з другої групи вважається синхронізованим та передається для подальшої обробки на наступний прохід виконання методу.

Під час синхронізації кожен кортеж часових значень  $\langle t_i^{(k)} \rangle_{i=1}^{n_k}$  розглядається як послідовність дискретних інтервалів  $\left[ t_\tau^{(k)} t_{\tau+1}^{(k)} \right]$ ,  $\tau = [1 \dots (n_k - 1)]$ .

Процедура синхронізації мультиобразів  $I_{p,I}$  та  $I_{p,II}$  включає такі дії:

1. Якщо  $\Delta_{p,I} = [t_1^{(p,I)} t_{n_{p,I}}^{(p,I)}]$  або  $\Delta_{p,II} = [t_1^{(p,II)} t_{n_{p,II}}^{(p,II)}]$  є неперіодичною послідовністю, то виконати синхронізацію даних за правилом УП1 та перейти на п. 11.
2. Інакше встановити інтервальне відношення дискретних інтервалів  $\Delta_{p,I} = [t_1^{(p,I)} t_{n_{p,I}}^{(p,I)}]$  та  $\Delta_{p,II} = [t_1^{(p,II)} t_{n_{p,II}}^{(p,II)}]$  та виконати синхронізацію даних відповідно до встановленого відношення:
  - 2.1. Якщо  $b_d(\Delta_{p,I}, \Delta_{p,II}) = true$ , то застосувати правило синхронізації мультимодальних даних БП61 та перейти на п. 11.
  - 2.2. Якщо  $bi_d(\Delta_{p,I}, \Delta_{p,II}) = true$ , то застосувати правило синхронізації мультимодальних даних БП62 та перейти на п. 11.
  - 2.3. Якщо  $md(\Delta_{p,I}, \Delta_{p,II}) = true$ , то застосувати правило синхронізації мультимодальних даних БП63 та перейти на п. 11.
  - 2.4. Якщо  $mi_d(\Delta_{p,I}, \Delta_{p,II}) = true$ , то застосувати правило синхронізації мультимодальних даних БП64 та перейти на п. 11.
  - 2.5. Якщо  $od(\Delta_{p,I}, \Delta_{p,II}) = true$ , то застосувати правило синхронізації мультимодальних даних БП65 та перейти на п. 11.
  - 2.6. Якщо  $oi_d(\Delta_{p,I}, \Delta_{p,II}) = true$ , то застосувати правило синхронізації мультимодальних даних БП66 та перейти на п. 11.
  - 2.7. Інакше перейти на п. 3.
3. Визначити дискретне значення  $t_1^{(p,I)}$ .
4. Якщо  $\exists t_\alpha^{(p,II)}$ , таке, що  $t_\alpha^{(p,II)} \leq t_1^{(p,I)}$ , то застосувати до  $[t_1^{(p,I)}]$  та  $[t_1^{(p,II)} t_\alpha^{(p,II)}]$  правило синхронізації БП61 ( $t_\alpha^{(p,II)} < t_1^{(p,I)}$ ) та/або правило синхронізації УП1 ( $t_\alpha^{(p,II)} = t_1^{(p,I)}$ ).

5. Якщо  $\exists t_{\beta}^{(p,II)}$ , таке, що  $t_{\beta}^{(p,II)} > t_2^{(p,I)}$ , то застосувати до  $\left[ t_1^{(p,I)} \right]$  та  $\left[ t_{\beta}^{(p,II)} t_{n_{p,II}}^{(p,II)} \right]$  правило синхронізації БП62.
6. Визначити дискретний інтервал  $\left[ t_1^{(p,I)} t_2^{(p,I)} \right]$ .
7. Визначити дискретний інтервал  $\left[ t_{\alpha}^{(p,II)} t_{\beta}^{(p,II)} \right]$  такий, що  $t_{\alpha}^{(p,II)} > t_1^{(p,I)}$  та  $t_{\beta}^{(p,II)} \leq t_2^{(p,I)}$ .
8. Встановити інтервальне відношення дискретних інтервалів  $\left[ t_1^{(p,I)} t_2^{(p,I)} \right]$  та  $\left[ t_{\alpha}^{(p,II)} t_{\beta}^{(p,II)} \right]$  та визначити базові правила синхронізації (якщо  $\Delta t = 0$ ) або правило нечіткої синхронізації (якщо  $\Delta t > 0$ ), умови яких виконуються для встановленого відношення.
9. Сформувати шаблон синхронізації дискретних інтервалів  $\left[ t_1^{(p,I)} t_2^{(p,I)} \right]$  та  $\left[ t_{\alpha}^{(p,II)} t_{\beta}^{(p,II)} \right]$ .
10. Відповідно до визначеного шаблону синхронізації виконати синхронізацію мультимодальних даних, що визначені для дискретних інтервалів  $\left[ t_{\tau}^{(p,I)} t_{\tau+1}^{(p,I)} \right]$  та  $\left[ t_{\alpha}^{(p,II)} t_{\beta}^{(p,II)} \right]$  для  $\forall \tau, 1 < \tau < n_{p,I}$  та  $\exists \alpha, \beta$ , таких, що  $t_{\alpha}^{(p,II)} > t_{\tau}^{(p,I)}$  та  $t_{\beta}^{(p,II)} \leq t_{\tau+1}^{(p,I)}$ .
11. Сформувати синхронізовані мультиобрази  $I_1^{S(P)}, \dots, I_K^{S(P)}$ .

Процедура синхронізації виконується для всіх  $\left\lfloor \frac{K}{2} \right\rfloor$  пар мультиобразів, після чого виконується наступний з  $P$  проходів методу.

Результатом виконання всіх проходів методу є синхронізовані мультиобрази  $I_1^S, \dots, I_K^S$ .

Виконати аналіз запропонованого методу можна лише якісно, оскільки інші методи синхронізації мультиобразів наразі не існують.

Розглянемо приклад застосування методу. Нехай потрібно синхронізувати дані двох мультиобразів, в яких часові значення – це години

однієї доби (періодична послідовність, похибка визначення часу  $\Delta t = 0$ ), а дані – це значення вологості ґрунту на певній ділянці (першій та другій) сільськогосподарських угідь, облаштованих за технологією Smart Agriculture [300]:

$$A = \llbracket T, M_h \mid \langle 6, 10, 14, 18, 22 \rangle, \langle 15.1, 28.0, 27.4, 25.9, 19.3 \rangle \rrbracket,$$

$$B = \llbracket T, M_h \mid \langle 6, 8, 10, 12, 14, 16, 18, 20, 22 \rangle, \langle 10.5, 21.0, 20.6, 20.1, 17.6, 15.4, 14.2, 12.0, 11.6 \rangle \rrbracket.$$

Для синхронізації цих мультиобразів потрібно виконання одного проходу методу ( $P = \lceil \log_2 2 \rceil = 1$ ). Довжина часового кортежу  $A$  складає  $n_A = 5$ , часового кортежу  $B$  складає  $n_B = 9$ . Оскільки  $n_A < n_B$  та  $P = 1$ , то єдина пара мультиобразів для синхронізації утворюється як  $\langle A, B \rangle$ .

Оскільки кортежі значень часу обох мультиобразів є періодичними послідовностями, то п. 1 процедури синхронізації не виконуємо та переходимо на п. 2 – встановлення виду інтервального відношення дискретних інтервалів  $\Delta_A = [t_1^A \ t_5^A] = [6 \ 22]$  та  $\Delta_B = [t_1^B \ t_9^B] = [6 \ 22]$ .

Оскільки умови пп. 2.1-2.6 не виконуються, то згідно з пп. 2.7 переходимо на п.3, згідно з ким визначаємо, що  $t_1^A = 6$ .

Далі у п. 4 визначаємо, що  $t_1^B = t_1^A$ , отже, застосовуємо правило УП1 до  $[t_1^A]$  та  $[t_1^B]$ , в результаті чого отримуємо  $d_1^A = 15.1$ ,  $d_1^B = 10.5$ .

Оскільки умова п. 5 не виконується, то переходимо на п. 6, де визначаємо дискретний інтервал  $]t_1^A \ t_2^A] = ]6 \ 10] = [10]$ .

Далі у п. 7 визначаємо, що  $\alpha = 2$ ,  $\beta = 3$ , тобто отримуємо дискретний інтервал  $[t_2^B \ t_3^B] = [8 \ 10]$ .

У п. 8 встановлюємо, що  $t_i^A = t_{i+1}^B$  та  $e_d([t_i^A], [t_i^B \ t_{i+1}^B]) = true$  при  $[t_i^A] \prec [t_i^B \ t_{i+1}^B]$ ,  $i = 2$ , тобто для синхронізації потрібно застосувати правило БП52, що визначається виразами (3.75).

Тоді у п. 9 шаблон визначаємо шаблон синхронізації БП52(2- $n_A$ ).

Далі у п. 10 застосовуємо створений шаблон до всіх дискретних інтервалів  $]t_\tau^A \ t_{\tau+1}^A]$  та  $[t_\alpha^B \ t_\beta^B]$ ,  $t_\alpha^B > t_\tau^A$  та  $t_\beta^B \leq t_{\tau+1}^A$ , де  $\tau = [2 \dots 4]$ :

$$\begin{array}{llllll} d_2^A = \emptyset & d_2^B = 21.0 & d_3^A = 28.0 & d_3^B = 20.6 & d_4^A = \emptyset & d_4^B = 20.1 \\ d_5^A = 27.4 & d_5^B = 17.6 & d_6^A = \emptyset & d_6^B = 15.4 & d_7^A = 25.9 & d_7^B = 14.2 \\ d_8^A = \emptyset & d_8^B = 12.0 & d_9^A = 19.3 & d_9^B = 11.6 & & \end{array}$$

За результатом виконання п. 11 процедури синхронізації отримуємо:

$$A^S = \llbracket T, M_h \mid \langle 6, 8, 10, 12, 14, 16, 18, 20, 22 \rangle, \langle 15.1, \emptyset, 28.0, \emptyset, 27.4, \emptyset, 25.9, \emptyset, 19.3 \rangle \rrbracket.$$

$$B^S = \llbracket T, M_h \mid \langle 6, 8, 10, 12, 14, 16, 18, 20, 22 \rangle, \langle 10.5, 21.0, 20.6, 20.1, 17.6, 15.4, 14.2, 12.0, 11.6 \rangle \rrbracket.$$

Завдяки застосуванню шаблону синхронізації кількість операцій над даними може бути зменшена у 2-5 разів порівняно із застосуванням виключно універсального правила синхронізації.

Отримані в результаті застосування методу створення мультиобrazу та методу синхронізації мультиобразів синхронізовані темпоральні мультимодальні дані є основою будь-якої моделі цифрового двійника досліджуваного об'єкта.

Розглянемо створення моделей цифрового двійника на основі концепції мультиобразу для різних випадків опису та відображення досліджуваного об'єкта програмним шляхом у вигляді його цифрового двійника.



### 3.4. Моделі цифрового двійника

Цифровий двійник подається у програмній системі через дані, що отримуються у процесі спостереження за фізичним двійником – досліджуваним об’єктом. Ці дані можуть розглядатись на різних рівнях абстракції, виходячи з мети дослідження фізичного двійника. Залежно від контексту конкретної вирішуваної задачі цифровий двійник може відображати фізичний двійник як:

- суцільний об’єкт, цей об’єкт може вважатись неподільним, але при цьому він може бути неоднорідним за своєю будовою, матеріалом, фізичними властивостями тощо;
- об’єкт, що визначається набором станів, динамічно змінюваних з плином часу;
- об’єкт, що є композицією своїх складових, кожна з яких в свою чергу може розглядатись як окремий об’єкт, утворюючи ієрархію об’єктів.

Ці рівні абстракції можуть комбінуватись для отримання більш інформативного способу опису фізичного двійника.

Для подання даних для кожного із зазначених рівнів абстракції потрібно розробити відповідні моделі, що дозволять якісне подання та ефективну обробку даних цифрового двійника досліджуваного об’єкта.

#### 3.4.1. Муксельна модель

Визначимо муксельну модель [5, 26], що дозволяє розглядати цифровий двійник досліджуваного об’єкта як суцільний об’єкт. Цей суцільний об’єкт визначається набором темпоральних мультимодальних даних, кожен з яких визначається певним мультиобразом.

Ідея муксельної моделі, ґрунтується на тому, що однією з важливих модальностей для візуального відображення цифрового двійника є

інформація про зовнішній вигляд його фізичного двійника – реального тривимірного об'єкта.

Найбільш ґрунтовну тривимірну графічну інформацію про реальні об'єкти дозволяють зберігати структури даних на основі вокселя (*voxel*, від *volume element*) – об'ємного елементу одиничного розміру [301–303]. Воксельна структура даних лежить в основі воксельної графіки. Воксель є розширенням поняття пікселя (*pixel*, від *picture element*) – одиничного елементу (точки) у растровій графіці. Так само, як і піксель, воксель характеризується деяким кольором. Воксельна графіка використовується для візуалізації медичної інформації (наприклад, об'ємна візуалізація результату МРТ-дослідження), для візуалізації наукових даних, у комп'ютерній анімації тощо. Подальшим розвитком та узагальненням понять «піксель» та «воксель» є доксель (*doxel*, від *dynamic element*) – воксель, визначений у часі [304, 305]. Ідею подання тривимірної графічної інформації, реалізовану у воксельній та доксельній графіці, доцільно поширити на збереження інформації інших модальностей, таких як аудіо-інформація, інформація про фізичні властивості та параметри об'єкта тощо.

Таким чином, мультимодальну інформацію про досліджуваний об'єкт можна зберігати на основі об'ємного елементу найбільш узагальненого типу, який зберігає інформацію всіх можливих модальностей, представлену у часі. За аналогією будемо називати такий елемент «муксель» (*mixel*, від *multimodal element*) [5, 26].

Тоді, модель цифрового двійника досліджуваного об'єкта може бути визначена через поняття мукселю як одиничного елементу досліджуваного об'єкта, який описується впорядкованою сукупністю темпоральних мультимодальних даних – мультиобразом.

Муксельна модель цифрового двійника досліджуваного об'єкта ґрунтується на декомпозиції простору об'єкта, як показано на рис. 3.7.

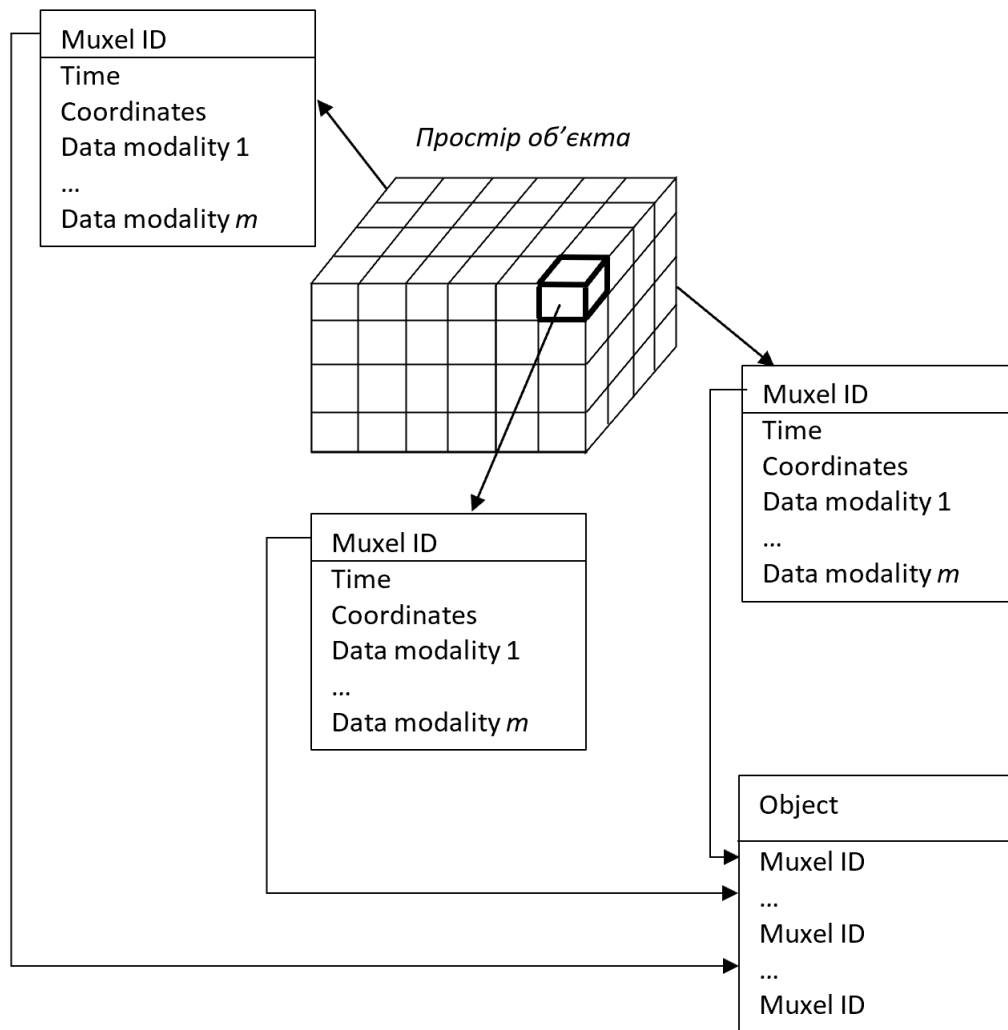


Рис. 3.7. Муксельна модель досліджуваного об'єкта

Створення муксельної моделі цифрового двійника досліджуваного об'єкта передбачає наступні етапи.

На першому етапі відбувається формування структури даних мультиобразу, який описує муксель досліджуваного об'єкта. Наприклад, для надання повного опису стану мукселя об'єкта структура даних має визначати такі дані:

- часові дані, що задають момент часу, у який фіксується мультимодальна інформація;
- просторові дані, що задають координати мукселя у просторі об'єкта;
- візуальні дані, що задають колір та прозорість відповідної цьому

- мукселю точки у просторі об'єкта;
- акустичні дані, задають значення характеристик акустичного сигналу у відповідних цьому мукселю точці простору;
  - фізичні дані, задають тип та стан речовини у відповідній цьому мукселю точці простору.

Для певних галузей застосування, може бути потрібно використання інших компонентів структури даних. Наприклад, для інженерних задач, що вирішуються у хімічній промисловості, може виникнути потреба в описі хімічних речовин. Їх формальний опис [306, 307] може бути застосований для визначення даних певної додаткової модальності у муксельній моделі.

В результаті виконання цього етапу отримуємо деяку формальну специфікацію мультиобразу. Приклад схематичної специфікації мультиобразу для опису мукселя показано на рис. 3.8.

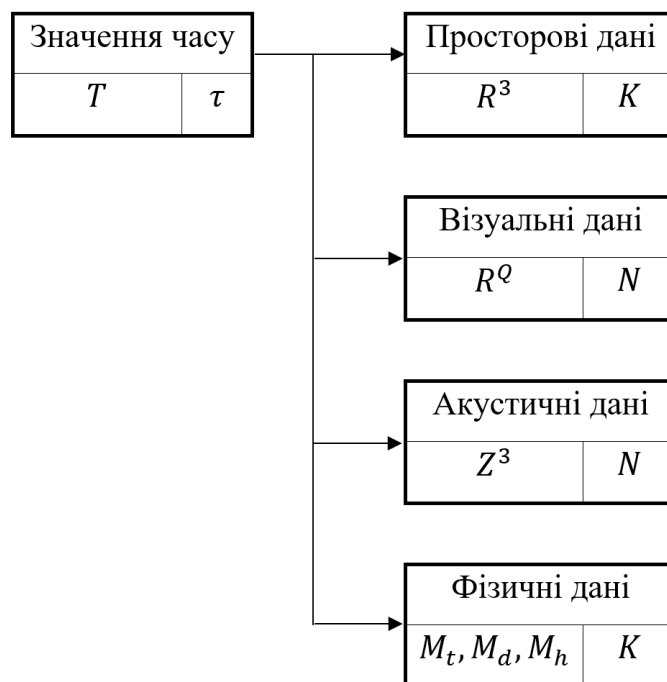


Рис. 3.8. Схематична специфікація мультиобразу для опису мукселя

Цій схематичній специфікації відповідає наступна математична специфікація:

$$I_{muxel} = \left[ \left[ T, R^3, R^Q, Z^3, \langle M_t, M_d, M_h \rangle | \langle t_i \rangle_{i=1}^\tau, \langle a_i^{(R^3)} \rangle_{i=1}^K, \langle a_i^{(R^Q)} \rangle_{i=1}^N, \right. \right. \\ \left. \left. \langle a_i^{(Z^3)} \rangle_{i=1}^N, \langle a_i^{(\langle M_t, M_d, M_h \rangle)} \rangle_{i=1}^K \right] \right],$$

де  $T$  – це множина значень часу;  $R^3$  – множина декартових координат точки простору об’єкта;  $R^Q$  – множина компонентів графічних даних;  $Z^3$  – множина значень акустичних компонентів;  $M_t$  – множина температурних даних;  $M_d$  – множина значень густини матеріалу;  $M_h$  – множина значень вологості матеріалу;  $\tau, N, K$  – кількість елементів кортежу даних.

На другому етапі відбувається формування структури об’єкта для визначення взаємозв’язку окремих мукселів суцільного об’єкта. Структура об’єкта може бути описана за допомогою масиву або дерева.

Визначення структури об’єкта на основі масиву є аналогічним визначенню форми тривимірного об’єкта у воксельній графіці [301–303].

Розіб’ємо простір об’єкта паралельними січними площинами у напрямку кожної з координатних осей системи координат цього простору. В результаті отримаємо набір елементарних паралелепіпедів, кожен з яких відповідає певному мукселю об’єкта. Таким чином, досліджуваний об’єкт може бути визначений як мультиобраз, що є результатом об’єднання мультиобразів мукселів, які утворюють цей об’єкт:

$$I_{object} = \bigcup_{z=0}^{d_z} \left( \bigcup_{y=0}^{d_y} \left( \bigcup_{x=0}^{d_x} I_{x,y,z} \right) \right), \quad (3.122)$$

де  $I_{x,y,z}$  – мультиобраз мукселя, що є асоційованим з точкою простору об'єкта, яка визначається координатами  $(x, y, z)$ ; точки з координатами  $(0,0,0)$  та  $(d_x, d_y, d_z)$  визначають діагональ паралелепіпеда, який щільно охоплює об'єкт (тобто є його осяжною оболонкою).

Точка простору, з якою не асоційований жоден муксель, описується мультиобразом, що є пустим агрегатом (Визначення 2.2).

Визначення структури об'єкта на основі дерева за принципом є схожим до визначення форми тривимірного об'єкта за допомогою розрідженого воксельного октодерева (Sparse Voxel Octree) [302], що застосовується для оптимізації воксельної моделі.

Для отримання дерева, що задає структуру об'єкта, розіб'ємо паралелепіпед, що є осяжною оболонкою об'єкта, навпіл січними площинами у напрямку кожної з координатних осей системи координат простору об'єкта.

Перевіримо наявність певної частини об'єкта у кожному з чотирьох паралелепіпедів, що утворились. Якщо у певному паралелепіпеді є частина об'єкта, знову розіб'ємо цей паралелепіпед навпіл у кожному напрямку та перевіримо наявність певної частини об'єкта у кожному з чотирьох паралелепіпедів, що утворились. Будемо рекурсивно повторювати поділ та перевірку, доки не досягнемо потрібного рівня деталізації або деякого мінімального об'єму простору, що асоціюється з одним мукселем.

На кожному кроці рекурсивного повернення будемо додавати вузли та гілки дерева об'єкта та асоціювати з вузлами мукселі та їх мультиобрази. Вузол, з яким не асоційований жоден муксель, описується мультиобразом, що є пустим агрегатом.

В результаті отримуємо структуру, яка графічно може бути визначена, як показано на рис. 3.9.

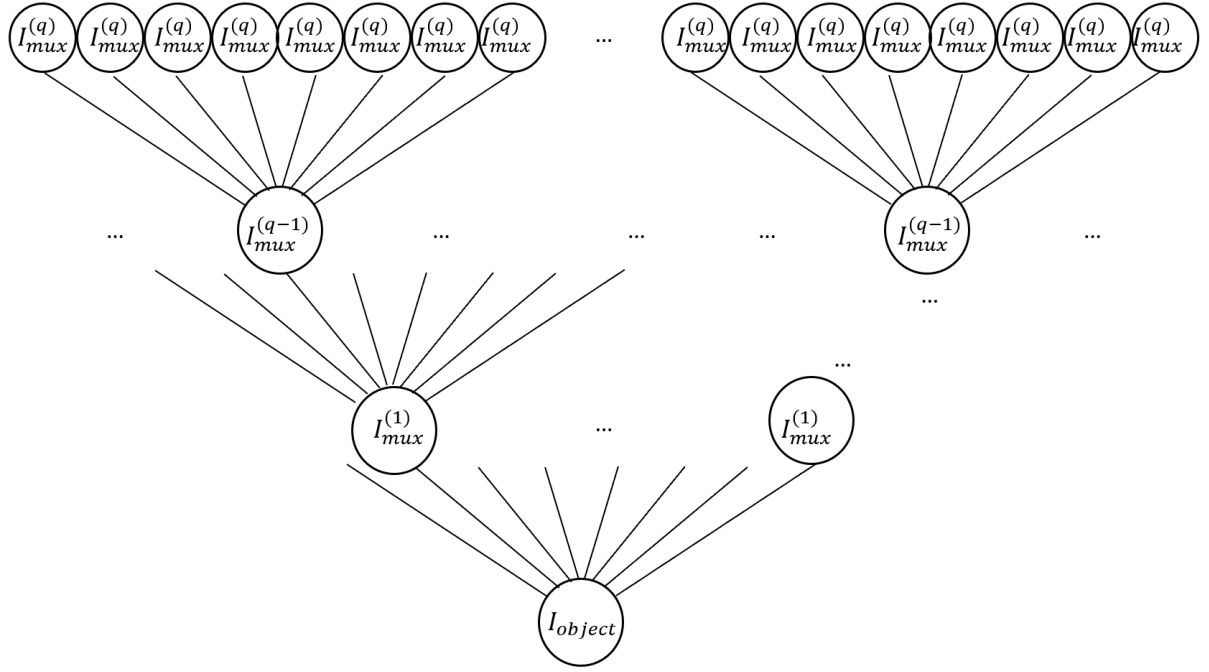


Рис. 3.9. Визначення структури об'єкта на основі октодерев

У цій структурі вузли найвищого рівня асоціюються з мультиобрами  $I_{mix}^{(q)}$  окремих мукселів:

$$I_{mix}^{(q)} = \left[ \left[ T, M_1, \dots, M_N | \langle t_i \rangle_{i=1}^{\tau}, \langle a_i^{(1)} \rangle_{i=1}^{n_1}, \dots, \langle a_i^{(N)} \rangle_{i=1}^{n_N} \right] \right] \quad (3.123)$$

На наступних рівнях вузли асоціюються з об'єднаннями мультиобразів попереднього рівня, тобто групами мукселів:

$$I_{mix}^{(q-1)} = \bigcup I_{mix}^{(q)} \quad (3.124)$$

Корень дерева асоціюється з мультиобразом усього об'єкта. Цей мультиобраз формується з мультиобразів першого рівня, тобто опис об'єкта включає опис всіх мукселів:

$$I_{object} = \bigcup I_{mux}^{(1)} \quad (3.125)$$

Кількість рівнів дерева визначається структурою фізичного двійника. Кожен вузол є зв'язаним з певним мультиобразом через ідентифікаційний код мукселя або групи мукселів.

Визначимо формат цього ідентифікаційного коду за допомогою нотації EBNF [299]:

*ідентифікаційний\_код* = *номер\_рівня*, “-”, *номер\_гілки*, “-”, *номер\_вузла*

Нумерація рівнів починається з нуля: нульовий рівень дерева – це його корінь. Оскільки дерево є октарним, то номер гілки та номер вузла у гілці є значеннями від 0 до 7. Корінь дерева має код 0-0-0.

Обхід дерева починається з кореня дерева та відбувається у прямому порядку [311], що визначається рекурсивною процедурою, наведеною у лістингу 3.1, в якій номер рівня позначимо як *i*, номер гілки позначимо як *j* та номер вузла позначимо як *k*.

Лістинг 3.1. Рекурсивна процедура обходу дерева структури об'єкта

```
PreOrder (i, j, k)
    IF поточний вузол існує
        обробка даних поточного вузла
        FOR (q = 0; q < 8; q++)
            PreOrder (i+1, k, q)
```

Отримане дерево однозначно визначає структуру об'єкта.

На третьому етапі відбувається отримання темпоральних мультимодальних даних та формування їх впорядкованої сукупності на основі структур, визначених на першому та другому етапах.



Залежно від властивостей досліджуваного об'єкта деякі елементи послідовності даних можуть не визначатись конкретним значенням, а замінятись особливими елементами-мітками:

- «Unknown» (UN), використовується, коли елемент даних певної модальності є невідомим – відповідає пустому елементу ( $\emptyset$ ) у математичній специфікації мультиобразу;
- «Not Applicable» (NA), використовується, коли елемент даних не визначається – відповідає невизначеному елементу ( $\square$ ) у математичній специфікації мультиобразу;
- «By Default» (BD), використовується для стандартних елементів даних, що задаються за замовчуванням; у цьому випадку відповідне значення береться з наперед визначеного набору стандартних елементів даних;
- «Repeated» (RP), використовується, коли значення повторюється для деякої послідовності мукселів або повторюється для одного мукселя для певного проміжку часу.

Кожен муксель може бути у чотирьох станах: інформаційному, невизначеному, квазівизначеному та у стані повторення.

Муксель у *інформаційному стані* визначається явно у всіх модальностях конкретними значеннями або елементами-мітками «By Default».

Муксель у *невизначеному стані* визначається елементами-мітками «Unknown» або «Not Applicable» у всіх модальностях. Цей стан означає, що дані про муксель або невідомі, або неважливі для даної задачі, тому не визначаються.

Муксель у *квазівизначеному стані* визначається явно лише у деяких модальностях, а у всіх інших задається мітками «Unknown».

Якщо дані всіх типів, окрім часових, певного мукселя збігаються з відповідними даними попереднього мукселя, то цей муксель може

визначатись, як такий, що знаходиться у *стані повторення*. В цьому випадку замість даних відповідної модальності задається мітка «Repeated». Стан повторення потрібен для ущільнення муксельних даних.

Частота фіксації зміни стану мукселів у часі має визначатись частотою зміни найбільш змінюваних даних, а також технічними можливостями програмно-апаратної системи.

Точність опису об'єкта визначається рівнем деталізації мукселя, вибір якого залежить від потрібної або можливої точності визначення мультимодальних даних.

Оскільки муксельна модель містить дані усіх модальностей про кожну точку простору об'єкта на деякому часовому проміжку, то обсяг цих даних є значним. Тому доцільно оптимізувати муксельну модель з точки зору способу подання даних. Це можна досягти ущільненням даних, застосуванням дерева як структури даних (замість масиву) або комбінацією першого та другого підходів.

Ущільнення даних муксельної моделі можливе, якщо частина мукселів знаходиться у невизначеному стані, квазівизначеному стані або у стані повтору. У цьому випадку компоненти сусідніх мукселів набувають однакового значення, в результаті чого утворюються довгі ланцюжки однакових елементів даних, що дозволяє ущільнення даних.

Для ущільнення даних муксельної моделі можуть бути застосовані методи ущільнення без втрат [308]. Зокрема, можуть застосовуватись методи словникового ущільнення (наприклад, метод LZW [309, 310]), методи групового кодування (наприклад, метод RLE [308]) та інші методи, які ефективно ущільнюють довгі послідовності однакових значень. Розглянемо застосування методу RLE.

Нехай послідовність мукселів знаходиться у стані повтору. Тоді отримуємо набір компонентів, наведений у лістингу 3.2.

### Лістинг 3.2. Визначення мукселя у стані повтору

```
Ідентифікатор_мукселя
{ значення_часу;
  Repeated;
  Repeated;
  Repeated;
  Repeated;
  Repeated }
```

Такі компоненти можуть бути замінені на пару виду: *<кількість\_повторень, екземпляр\_компоненту>*. Тоді екземпляр компоненту може бути записаний, як показано у лістингу 3.3.

### Лістинг 3.3. Компактне визначення мукселя у стані повтору

```
Ідентифікатор_мукселя
{ значення_часу;
  <5, Repeated> }
```

Дані муксельної моделі зберігаються у наборі файлів таких типів:

1. Файл структури об'єкта.
2. Файл структури даних мультиобраза.
3. Файл стандартних даних.
4. Файл даних певної модальності.

Формат файлу структури об'єкта залежить від способу визначення структури об'єкта.

Якщо структура об'єкта визначена як масив, то може бути застосований формат файлу, що наведений на рис. 3.10. Цей формат ґрунтується на форматі CSV [312], проте він передбачає застосування кількох типів розділювачів значень.

У секції метаданих визначається тип структури – матриця (MATRIX). Далі зазначаються розміри простору об'єкта, який задається осяжною

оболонкою – паралелепіпедом розміром  $x \times y \times z$  мукселів. Далі зазначається ім'я файлу, в якому задано структуру даних мультимедіа, що визначає муксель.

У секції даних впорядковано розташовуються ідентифікатори мукселів. Окремі ідентифікатори розділяються комою, рядки з  $x$  елементів (ідентифікаторів) поділяються крапкою з комою, матриці елементів розміром  $x \times y$  завершуються знаком #. Файл закінчується маркером EOF.

Metadata Section
"MATRIX" (6 bytes) Matrix dimensions x, y, z (6 bytes) Multi-image structure file name (128 bytes) Standard data tables file name (128 bytes)
Data Section
Muxel 1-1-1 ID, Muxel 2-1-1 ID, ..., Muxel x-1- <u>1</u> ; Muxel 1-2-1 ID, Muxel 2-2-1 ID, ..., Muxel x-2- <u>1</u> ; ... Muxel 1-y-1 ID, Muxel 2-y-1 ID, ..., Muxel x-y-1#
...
Muxel 1-1-z ID, Muxel 2-1-z ID, ..., Muxel x-1- <u>z</u> ; Muxel 1-2-z ID, Muxel 2-2-z ID, ..., Muxel x-2- <u>z</u> ; ... Muxel 1-y-z ID, Muxel 2-y-z ID, ..., Muxel x-y-z#

Рис. 3.10. Формат файлу структури об'єкта (матриця)

Якщо структура об'єкта визначена як дерево, то передбачається застосування формату (рис. 3.11), що побудований за принципом, який

лежить в основі формату JSON [134].

У секції метаданих визначається тип структури – матриця (OCTREE). Далі зазначається кількість рівнів. Далі зазначається ім'я файлу, в якому задано структуру даних мультимедіа, що визначає муксель.

Metadata Section
<p>“OCTREE” (6 bytes)</p> <p>Levels quantity (6 bytes)</p> <p>Multi-image structure file name (128 bytes)</p> <p>Standard data tables file name (128 bytes)</p>
Data Section
<p>“Level 0” Root ID (0-0-0)</p> <p>{</p> <p>    “Level 1” {</p> <p>        Node ID, Node ID, Node ID, Node ID,</p> <p>        Node ID, Node ID, Node ID, Node ID }</p> <p>    “Level 2” {</p> <p>        Node ID, Node ID, Node ID, Node ID,</p> <p>        ...</p> <p>        Node ID, Node ID, Node ID, Node ID }</p> <p>    .....</p> <p>    “Level N” {</p> <p>        Node ID, Node ID, Node ID, Node ID,</p> <p>        ...</p> <p>        Node ID, Node ID, Node ID, Node ID }</p> <p>}</p>

Рис. 3.11. Формат файлу структури об'єкта (дерево)

У секції даних впорядковано розташовуються ідентифікатори мукселів, формат передбачає ієрархічне подання дерева: рівень за рівнем. Кожен рівень позначається знаками { i }. Файл закінчується маркером EOF.

Файл структури даних мультиобраза ґрунтується на форматі JSON [134] та має вигляд, наведений на рис. 3.12.

Metadata Section
“MULTI-IMAGE” (11 bytes) Aggregate length (4 bytes)
Sets Section
Set 1 { “Name” : set name; “Type” : type description } ..... Set N { “Name” : set name; “Type” : type description }
Tuples Section
Tuple 1 { “Name” : tuple name; “Set” : set name; “Length” : number of elements } ..... Tuple N { “Name” : tuple name; “Set” : set name; “Length” : number of elements }

Рис. 3.12. Формат файлу структури даних мультиобраза

У секції метаданих зазначається ознака типу файлу (MULTI-IMAGE). Далі задається кількість кортежів у мультиобразі. У секції множин задається тип даних для кожного кортежу. Перша множина визначає формат значень часу. Тип даних може задаватись як ієрархічна структура. Для визначення її рівнів використовуються знаки { та }. У секції кортежів визначаються кортежі кожної модальності – їх довжини та тип елементів. Першим кортежем є кортеж значень часу.

Формат файлу стандартних даних передбачає структуру, що наведена на рис. 3.13.

Metadata Section
“STANDARD_DATA” (13 bytes) Tables quantity (4 bytes)
Tables Section
Table 1 { “Name” : tuple name; “Data format” : format definition; “Table length” : number of elements; “Data” { structured data } } ..... Table N { “Name” : tuple name; “Data format” : format definition; “Table length” : number of elements; “Data” { structured data } }

Рис. 3.13. Формат файлу стандартних даних

У секції метаданих зазначається ознака типу файлу (STANDARD\_DATA). Далі зазначається кількість таблиць стандартних

даних. Секція таблиць містить опис кожної таблиці (назва, формат даних, кількість елементів) та її впорядковані дані. Прикладами стандартних таблиць даних моделі є:

- таблиця часових міток;
- таблиця кольорів;
- таблиця матеріалів;
- таблицю станів тощо.

Формат файлу даних кожної модальності визначається залежно від програмного забезпечення, яке може бути використано для обробки та подання даних цієї модельності. Наприклад, файл тривимірних графічних даних може мати тип STL [131], OBJ [313], VRML [314] тощо, файл аудіо даних може мати тип MP3 [315], FLAC [316], OGG [317] тощо, файл температурних даних може мати тип CSV [312].

Пропонуються наступні розширення для файлів, що визначають муксельну модель:

- файл структури об'єкта – .osr;
- файл структури даних мультиобраза – .mos;
- файл стандартних даних – .sta.

Сукупність файлів муксельної моделі зберігається у окремій теці.

Муксельна модель може бути порівняна з узагальненою воксельною моделлю [303], яку було створено для розширення можливостей 3D візуалізації томографічних зображень органів людського тіла. Узагальнена воксельна модель дозволяє зберігати, крім графічних даних вокселів, також метадані – ознаку приналежності вокселя певному органу. На відміну від узагальненої воксельної моделі, муксельна модель передбачає збереження темпоральних мультимодальних даних, що дозволяє застосовувати її для опису досліджуваного об'єкта довільного типу.

Визначимо деякі можливості, які надає муксельна модель. Оскільки муксельна модель містить повну інформацію про об'єкт, то ця модель



робить можливим «проекціювання» інформації кожної модальності відносно потрібної позиції спостереження та отримання зрізів цієї інформації, подібно до отримання зрізів воксельної моделі [15]. Наприклад, вона дозволяє отримати статичне растрове зображення під певним кутом погляду, визначити рівень акустичного сигналу у певній точці простору з урахуванням наявних перешкод на шляху розповсюдження акустичних хвиль, визначити наявність та інтенсивність певної властивості речовини, з якої складається об'єкт або його компоненти тощо.

Муксельну модель доцільно використовувати, коли є потреба у поданні деталізованої інформації для всебічного опису невеликого об'єкта.

### **3.4.2. Часова зв'язна модель**

Часова зв'язна модель [26] подає досліджуваний об'єкт через сукупність його дискретних станів, що визначаються в унікальні моменти часу набором характеристик об'єкта, які являють собою мультимодальні дані (рис. 3.8). У кожен момент часу у об'єкта є один та лише один стан, який визначається його мультиобразом. При цьому часовий кортеж кожного мультиобразу містить лише одне значення, яке визначає момент часу, в який об'єкт мав певний конкретний стан. Такий підхід дозволяє використовувати часове значення як *ключ*, що однозначно визначає цей конкретний стан у впорядкованій послідовності станів досліджуваного об'єкта (рис. 3.14).

Для створення часової зв'язної моделі потрібно визначити структуру мультиобразу досліджуваного об'єкта. При цьому мультиобраз для часової зв'язної моделі відрізняється тим, що його часовий кортеж містить лише одне значення часу, яке асоціюється з даними інших модельностей, що визначають стан досліджуваного об'єкта.

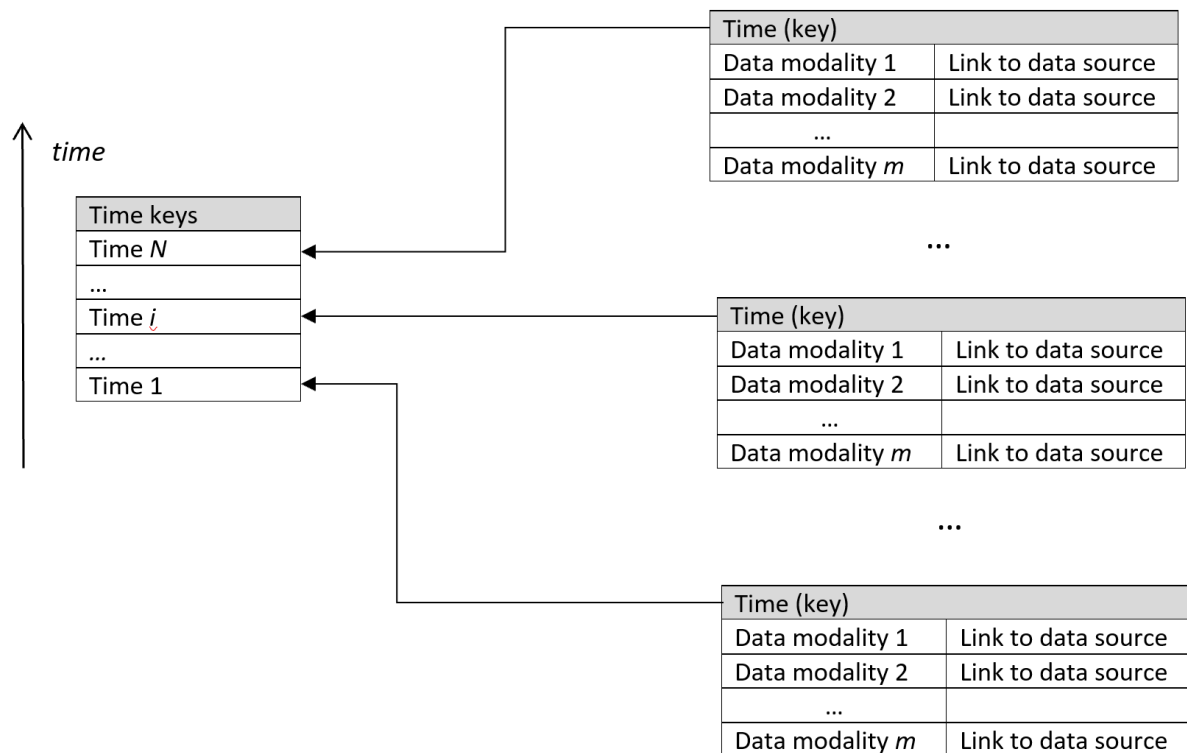


Рис. 3.14. Часова зв'язна модель досліджуваного об'єкта

Для того, щоб сформувати стан об'єкта у певний момент часу потрібно визначити джерела даних кожної модальності, що визначають цей стан. Це відбувається шляхом призначення унікальних ідентифікаторів, що являють собою посилення (зв'язки, *links*). Якщо дані певної модальності у поточний момент часу не можуть бути отримані чи не потрібні, то набору даних замість ідентифікатора зв'язку призначається атрибут *unavailable*.

Стани об'єкта є дискретними, це означає, що на уявній часовій шкалі опису об'єкта визначаються окремі дискретні моменти часу (часові ключі, *time keys*), з якими асоціюється конкретні стани об'єкта. При визначені кожного дискретного стану дані, що надійшли або почали надходити у певний дискретний момент часу, який є ключем цього поточного часу, синхронізуються та агрегуються.

Таким чином, від кожного джерела отримуємо дані та формуємо частковий мультиобраз наступного виду:

$$I_j = \llbracket T, M_j | \langle t^j \rangle, \langle d_i^j \rangle_{i=1}^{n_j} \rrbracket, \quad (3.126)$$

де  $t^j$  – часовий ключ  $j$ -го стану досліджуваного об’єкта,  $t^j \in T$ ;

$d_i^j$  – дані  $j$ -ї модальності, що асоційовані (визначені) з ключем  $t^j$ ,  $d_i^j \in M_j$ ;

$T$  – множина часових значень;

$M_j$  – множина даних  $j$ -ї модальності,  $j = [1 \dots N]$ .

Часткові мультиобрази об’єднуються в єдиний мультиобраз, що визначає поточний стан досліджуваного об’єкта, при цьому під час синхронізації та агрегації даних часткових мультиобразів серед всіх часових ключів обирається один часовий ключ, який і визначатиме цей поточний стан об’єкта, а сама процедура синхронізації відбувається згідно з правилами нечіткої синхронізації. В результаті, для визначення стану досліджуваного об’єкта отримуємо мультиобраз, часовий кортеж якого містить одне єдине значення:

$$I = \cup_{j=1}^N I_j = \llbracket T, M_1, \dots, M_N | \langle t \rangle, \langle d_i^1 \rangle_{i=1}^{n_1}, \dots, \langle d_i^N \rangle_{i=1}^{n_N} \rrbracket, \quad (3.127)$$

де  $\langle t \rangle$  визначається (3.128), (3.129), (3.130) або семантично.

$$\langle t \rangle = \max_{j=1 \dots N} t^j \quad (3.128)$$

$$\langle t \rangle = \min_{j=1 \dots N} t^j, \quad (3.129)$$

$$\langle t \rangle = \frac{\sum_{j=1}^N t^j}{N}, \quad (3.130)$$

Формула (3.128) означає, що за часовий ключ стану береться момент часу, коли всю сукупність даних отримано. Формула (3.129) означає, за ключ береться момент часу, коли дані тільки почали надходити. Формула (3.130) визначає усереднений час надходження даних.

Джерелом даних може бути давач, що генерує поточкові дані, або сховище (локальне, хмарне), що зберігає дані у вигляді файлів.

Розглянемо випадок, коли послідовність даних  $\langle d_i^j \rangle_{i=1}^{n_j}$  збережено у вигляді файлу певного формату. В деяких форматах файлів (наприклад, у форматах медичних зображень NIFTI [318] та DICOM [319]) час отримання компонентів цієї послідовності даних зберігається у метаданих або безпосередньо разом з основними даними.

Проте, у багатьох випадках темпоральна інформація, що стосується послідовності даних обмежується часом створення (модифікації) файлу, а інформація про час отримання окремих компонентів послідовності даних зберігається окремо. Тому задача синхронізації даних різних модальностей, що має бути виконана при створенні часової зв'язної моделі, може бути ускладнена через відсутність єдиного способу зберігання часових значень.

Щоб спростити обробку темпоральних мультимодальних даних при створенні часової зв'язної моделі, доцільно модифікувати файли, у яких зберігаються дані кожної з модальностей, що використовуються у моделі, за допомогою файлу-обгортки.

Структура цього файлу (рис. 3.15) включає три розділи: розділ метаданих, розділ часових значень та головний розділ, де знаходиться обгорнутий файл.

Розділ з метаданими містить маркер "TIME", ідентифікатор типу файлу (наприклад, "nii" для файлового формату NIFTI, "dcm" для формату файлу DICOM тощо) та розмір кортежу значень часу. Розділ часових значень містить значення часу, впорядковані за зростанням.

Metadata section
"TIME" (4 bytes) Wrapped file type (4 bytes) Length of time tuple (4 bytes)
Time stamps section
Time values tuple (N bytes)
Main section (wrapped data file)
Original data file (N*m)

Рис. 3.15. Структура файлу-обгортки

Основний розділ містить обгорнутий файл; цей файл є готовим до використання даних, наприклад, для їх відтворення або обробки програмним шляхом. Така процедура модифікації оригінального файлу дозволяє уніфікувати процедуру синхронізації послідовностей даних, отриманих з джерел різних типів, та при цьому зберегти дані у їх оригінальному форматі.

Часову зв'язну модель цифрового двійника доцільно використовувати у випадках, коли дослідження фізичного двійника відбувається через визначення його характеристик у дискретні моменти часу, в результаті чого відбувається формування послідовності дискретних станів досліджуваного об'єкта, кожен з яких характеризується сукупністю темпоральних мультимодальних даних, асоційованих з конкретним моментом часу, що може слугувати часовим ключем певного стану об'єкта.

### 3.4.3. Багаторівнева онтологічна модель

У багаторівневій онтологічній моделі [26] досліджуваний об'єкт подається як композиція його складових (компонентів). Кожен компонент об'єкта характеризується певними мультисемедійними параметрами та поведінковими характеристиками, які визначаються відповідними мультимодальними даними. Наприклад, двигун автомобіля являє собою складну технічну систему, яка складається з набору деталей. Кожна деталь має певний зовнішній вигляд та параметри (наприклад, геометричний розмір, тип матеріалу тощо), що відображається за допомогою візуальної моделі цифрового двійника двигуна, та характеризується певними характеристиками, які відтворюються за допомогою поведінкової моделі та можуть бути статичними (наприклад, міцність) або динамічно-змінюваними (швидкість обертання, частота вібрації тощо).

Компоненти багаторівневої онтологічної моделі пов'язані семантично, оскільки вони разом утворюють певну систему, якою є досліджуваний об'єкт. Це є макрорівень багаторівневої онтологічної моделі. Оскільки компоненти моделі є різними за своєю функціональністю, призначенням, будовою тощо, то для їх опису потрібна певна деталізація, що визначає мікрорівень багаторівневої онтологічної моделі. Ця деталізація може бути забезпечена шляхом застосування муксельної моделі, часової зв'язної моделі або будь якого іншого способу визначення конкретного компонента об'єкта. Таким чином, багаторівнева онтологічна модель є найвищим рівнем абстракції цифрового двійника досліджуваного об'єкта.

Деталізація потрібна лише для даних деяких модальностей, таких як, наприклад, аудіовізуальні дані, в той час, як для інших даних (наприклад, швидкість обертання шківу) вона не має сенсу.

Багаторівнева онтологічна модель визначається як орієнтований граф, в якому кожен вузол описується набором мультимодальних даних

(рис. 3.16). Зв'язки між вузлами показують семантичну підпорядкованість компонентів один одному. Кожен компонент визначається певним специфічним мультиобразом, тобто у загальному випадку мультиобрази компонентів відрізняються один від одного як за складом, так і за типом та кількістю елементів.

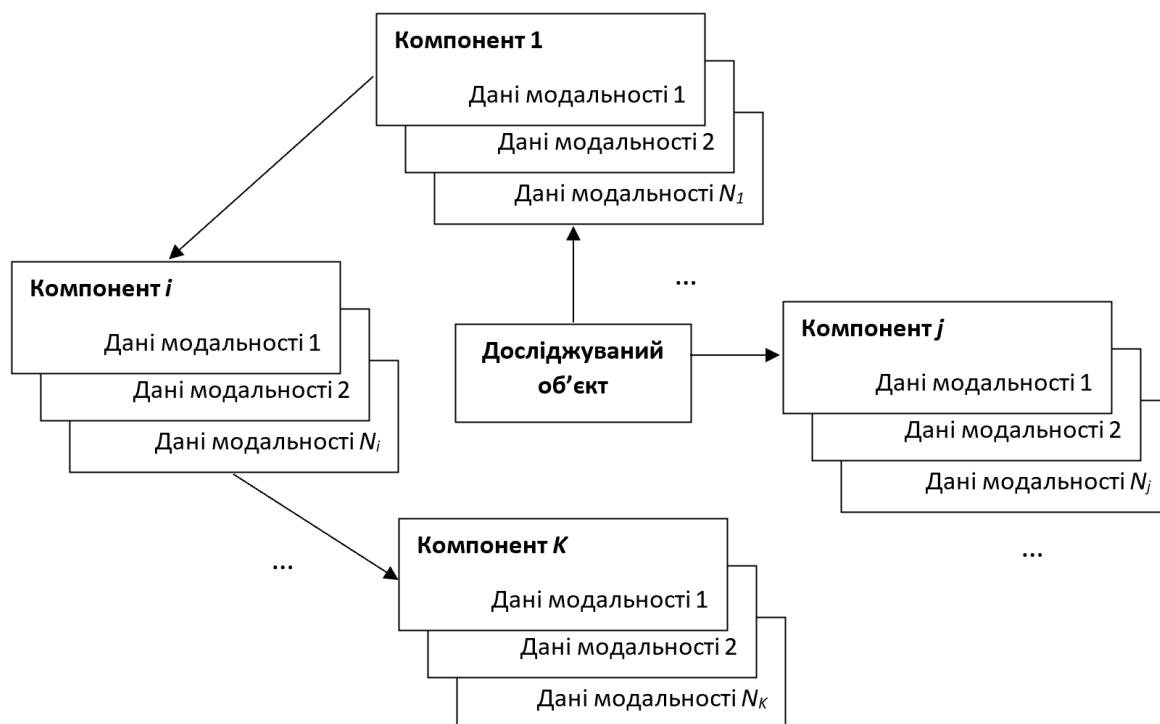


Рис. 3.16. Багаторівнева онтологічна модель досліджуваного об'єкта

У багаторівневій онтологічній моделі простір об'єкта може бути розширений до простору сцени, що містить декілька об'єктів. Тоді маємо вкладені багаторівневі онтологічні моделі та більшу кількість рівнів. Макрорівень такої вкладеної багаторівневої онтологічної моделі визначає сцену, яка включає об'єкти в цілому, а мікрорівень вкладеної багаторівневої онтологічної моделі визначає кожен об'єкт окремо.

Багаторівнева онтологічна модель на макрорівні може реалізована у вигляді графової бази даних (Graph Database), наприклад, Neo4j [320] та Amazon Neptune [321], оскільки сховище такого типу дозволяє збереження

взаємозв'язків та навігації по ним. Вузли у графовій базі даних використовуються для зберігання сутностей даних, а ребра – для зберігання взаємозв'язків між сутностями.

Реалізація мікрорівня в разі застосування для збереження деталізованих даних муксельної моделі може бути виконана на основі сховища об'єктів (Object Data Stores), оскільки база даних такого типу призначена для збереження аудіо- та відео-потоків, зображень та інших файлів великого обсягу. Прикладами сховища об'єктів є Azure Data Lake Store [322] та Azure File Storage [323].

Якщо для збереження деталізованих даних використовується часова зв'язна модель, то реалізація мікрорівня може бути виконана з допомогою сховища даних часових рядів (Time Series Data Stores), такого, як Prometheus [324] та TimescaleDB [325], що призначене для збереження даних часових рядів, зокрема, даних телеметрії.

Одним з можливих призначень багаторівневої онтологічної моделі є використання у інформаційно-пошукових системах. Оскільки багаторівнева онтологічна модель містить семантично-пов'язані мультимодальні дані, то її доцільно використовувати як інформаційну базу для пошуку подібних об'єктів (наприклад, знаходження всіх двигунів, які мають циліндр певного розміру). В разі такого застосування багаторівневої онтологічної моделі, вона може бути побудована як сховище даних зовнішніх індексів (External Index Data Stores), наприклад, за допомогою Azure Search [326], оскільки таке сховище дозволяє шукати дані, що зберігаються у інших сховищах даних, якими у цьому випадку будуть сховища даних мікрорівня, які реалізують муксельну та часову зв'язну моделі.

Основним призначенням багаторівневої онтологічної моделі є опис структури та деталізоване відображення окремих компонентів цифрового двійника досліджуваного об'єкта.



### 3.5. Метод адаптивного ущільнення графічних даних цифрового двійника

Графічні дані є одними з найбільш надлишкових серед усіх даних цифрового двійника. Тому доцільно виконувати їх ущільнення для зменшення обсягу даних, що підлягають зберіганню. Проте, при застосуванні алгоритмів ущільнення з втратами [32, 44], які дозволяють досягти найвищого ступеню ущільнення, виникає ризик зниження якості зображення після його відновлення з ущільнених графічних даних. Тому доцільно брати до уваги структуру зображення шляхом визначення *областей інтересу*. Сформулюємо метод адаптивного ущільнення графічних даних [30], який дозволяє визначати області інтересу на зображенні та обминати ці області під час ущільнення з метою уникнення втрати якості візуалізації цих областей при відтворенні зображення.

Довільне зображення  $A_a$  може бути розглянуто як відкрита множина, така, що:

$$\left\{ a_{ij} \in A_A \left| \begin{array}{l} 0 \leq a_{ij} \leq L \\ 1 \leq i \leq n \\ 1 \leq j \leq m \end{array} \right. \right\}, \quad (3.131)$$

де  $L$  – кількість градацій відтінків кольору,  
 $m$  та  $n$  задають розмір зображення у пікселях.

Цифрове зображення  $A$  розміру  $m \times n$  пікселів, отримане в результаті аналого-цифрового перетворення зображення  $A_a$ , може бути описане в матричному вигляді:

$$A = \begin{bmatrix} a_{11} & \dots & a_{1m} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nm} \end{bmatrix}, \quad (3.132)$$

де  $a_{ij}$  – числове значення відтінку кольору  $ij$ -пікселя.

З точки зору інформаційної цінності виділимо у зображенні якнайменше одну область  $G$ , яку називатимемо *областю інтересу*. З математичної точки зору область інтересу  $G$  являє собою зв'язну відкриту множину в евклідовому просторі, яке задане своєю границею.

*Маскою зображення  $A$*  називатимемо матрицю  $T$  розмірності  $m \times n$ , причому  $t_{ij} \in \{0,1\}$ , де  $t_{ij} = 1$ , якщо  $ij$ -піксель належить області інтересу  $G$ , інакше  $t_{ij} = 0$ .

Тоді задача полягає у побудові такої маски  $T$ , яка максимально точно відображає положення області інтересу  $G$  на заданому зображенні  $A$ . В термінах теорії множин маска  $T$  являє собою замикання множини  $G$  на заданому просторі зображення  $A$ .

Спосіб побудови маски  $T$  для заданого зображення  $A$  залежить від того, чи є область інтересу  $G$  заданою, чи її необхідно знайти, виходячи з деякого критерія.

Нехай область інтересу  $G$  задана. Область інтересу  $G$  може бути прямокутної та довільної форми. Якщо область інтересу має прямокутну форму, то для її завдання достатньо визначити номери рядка та стовпця двох пікселів,  $a_{kl}$  та  $a_{pq}$ , зображення  $A$ , що розташовані у протилежних кутах прямокутника області інтересу  $G$  ( $k < p, l < q$ ). В цьому випадку задача побудови маски  $T$  є тривіальною, оскільки зводиться до заповнення матриці  $T$ , що спочатку є нульовою, одиницями у позиціях  $t_{ij}$ , де  $k \leq i \leq p$ ,  $l \leq j \leq q$ .

Нехай область інтересу має довільну форму, тобто задана своєю границею. Поділимо множину пікселів зображення  $A$  на дві підмножини: підмножину пікселів, які належать області інтересу  $G$  та підмножину пікселів, які їй не належать. Такий поділ можливий, оскільки за умовою

розташування границі області інтересу  $G$  на зображенні є відомим. В цьому випадку потрібно заповнити матрицю  $T$ , що спочатку є нульовою, одиницями у позиціях  $t_{ij}$ , якщо  $a_{ij} \in G$ .

Тепер розглянемо визначення області інтересу, виходячи з деякого критерія. Зокрема, розглянемо два випадки:

1. Область інтересу визначається, виходячи з структурних особливостей зображення;
2. Область інтересу визначається, виходячи з границь об'єктів, присутніх у зображенні.

У першому випадку під структурними особливостями розуміється неоднорідність зображення з точки зору наявності в ньому так званих областей сталості, де всі пікселі мають однаковий або близький за значенням відтінки кольору, і областей з великою кількістю дрібних деталей, де сусідні пікселі істотно відрізняються за кольором. Для виділення цих областей можна скористатися статистичними характеристиками, що визначаються за гістограми яскравості. Зокрема можна проводити аналіз зображення на основі значення середньої ентропії [327, 328]:

$$e = - \sum_{i=0}^{L-1} g(b_i) \log_2 g(b_i), \quad (3.133)$$

де  $L$  – кількість градацій відтінків кольору,

$g(b_i)$  – гістограма яскравості,

$b$  – випадкова величина, що відповідає яскравості елементів зображення.

Ентропія характеризує мінливість яскравості зображення: вона набуває максимального значення в разі рівномірних значень, тобто за наявності великої кількості дрібних деталей, та  $e = 0$  для областей сталості.

Тоді визначимо наступний спосіб отримання маски.

Обчислимо значення ентропії  $e_0$  для всього зображення  $A$ . Далі розділимо зображення на  $N$  областей  $A_i^1$ , що не перетинаються, так, що:

$$\begin{cases} A_1^1 \cup A_2^1 \cup \dots \cup A_N^1 = A \\ A_1^1 \cap A_2^1 \cap \dots \cap A_N^1 = \emptyset \end{cases} \quad (3.134)$$

Нехай, наприклад,  $N = 4$ , тоді маємо 4 квадранти матриці зображення  $A$ .

Обчислимо значення ентропії  $e_i^1$  для кожної області  $A_i^1$  та порівняємо отримані значення зі значенням ентропії  $e^0$  для всього зображення  $A$ . Ознакою наявності дрібних деталей в області є збільшення значення ентропії в порівнянні з ентропією всього зображення. Таку область (одну або декілька) першого рівня розділимо на області  $A_i^2$  другого рівня так, що:

$$\begin{cases} A_1^2 \cup A_2^2 \cup \dots \cup A_N^2 = A^1 \\ A_1^2 \cap A_2^2 \cap \dots \cap A_N^2 = \emptyset \end{cases} \quad (3.135)$$

та обчислимо для кожної з областей другого рівня значення ентропії  $e_i^2$ . Ознакою наявності дрібних деталей в області другого рівня є збільшення значення ентропії у порівнянні з ентропією відповідно області першого рівня.

Вказану процедуру потрібно продовжувати до досягнення заданого рівня деталізації або до досягнення мінімального розміру  $2 \times 2$  області пікселя.

Маска  $T$ , що спочатку є нульовою матрицею, заповнюється одиницями в позиціях, які відповідають пікселям зображення  $A$ , що належить областям з максимальною ентропією.

У другому випадку, коли область інтересу визначається, виходячи з меж об'єктів, присутніх у зображенні, виділення границь областей

зображення може бути виконане з використанням операцій морфологічної обробки зображення: дилатації, ерозії, центрального відображення та паралельного перенесення [328]. Розглянемо їх докладніше.

Нехай  $D$  та  $H$  – множини у просторі  $Z^2$ , елементами яких є координати пікселів, що подають об'єкти на зображенні.

Центральне відображення множини  $D$  визначається як:

$$\hat{D} = \{\hat{d} \mid \hat{d} = -d, d \in D\}. \quad (3.136)$$

Паралельне перенесення множини  $H$  у точку  $z = (z_1, z_2)$  визначається за правилом:

$$(H)_z = \{\hat{h} \mid \hat{h} = h + z, h \in H\}. \quad (3.137)$$

Дилатація множини  $D$  на множині  $H$  визначається як:

$$D \oplus H = \{z \mid (H)_z \cap D \neq \emptyset\}. \quad (3.138)$$

Ерозія множини  $D$  на множині  $H$  визначається як:

$$D \ominus H = \{z \mid (H)_z \subseteq D\}. \quad (3.139)$$

Тоді границя  $\beta(D)$  множини  $D$  може бути виділена шляхом виконання ерозії множини  $D$  на множині  $H$  з подальшим отриманням різниці множини  $D$  та результату його ерозії:

$$\beta(D) = D \setminus (D \ominus H). \quad (3.140)$$

Матрицю  $H$  називатимемо примітивом. Може бути використаний примітив виду:

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (3.141)$$

Тоді побудова маски  $T$  полягає у заповненні матриці  $T$ , що спочатку є нульовою, одиницями в позиціях, що належать областям, які знаходяться всередині знайдених кордонів.

Процедура заповнення області в межах виявленої границі полягає в тому, щоб, почавши з деякого пікселя всередині цієї границі, заповнити одиничними значеннями всю область. При цьому передбачається, що границя є замкненою. Якщо вона не є замкненою, то її можна замкнути, з'єднавши крайні пікселі границі шляхом заповнення одиницями позицій матриці, що знаходяться між цими пікселями. До заповнення області одиницями призведе наступна рекурентна процедура:

$$t_{ij} = (t_{ij-1} \oplus H) \cap D^c, \quad (3.142)$$

де  $i$  та  $j$  визначають позицію всередині області,

$D^c$  – доповнення множини  $D$ ,

$H$  – примітив виду (3.143).

$$H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \quad (3.143)$$

В результаті виконання зазначеної рекурентної процедури отримуємо маску  $T$ , заповнену одиницями в позиціях, що відповідають об'єктам на

зображенні  $A$ .

Маска  $T$  являє собою бінарну матрицю. Для її компактного представлення з метою зберігання з іншими даними про зображення необхідно ущільнити матрицю, тобто представити у вигляді, коли для зберігання маски потрібно мінімальну кількість байтів.

Пропонується спосіб ущільнення маски, який полягає в перетворенні матриці  $T$  на вектор та записі його у вигляді послідовності десяткових чисел, кожне з яких представляє довжину групи однакових елементів, що стоять поспіль в отриманому бінарному векторі.

Ущільнення маски  $T$  в цьому випадку полягає у перетворенні матриці  $T$  на вектор таким чином, щоб елементи вектору утворювали групи однакових елементів, що стоять поспіль, максимально можливої довжини. Розглянемо це на прикладі. Нехай дана матриця  $X$  виду:

$$X = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Матриця  $X$  може бути перетворена на вектор різними способами:

- шляхом послідовного запису стовпців,
- шляхом послідовного запису рядків,
- шляхом послідовного запису елементів, розташованих в особливому порядку, наприклад, в порядку, що задається матрицею  $Z$ :

$$Z = \begin{bmatrix} 1 & 2 & 6 & 7 \\ 3 & 5 & 8 & 13 \\ 4 & 9 & 12 & 14 \\ 10 & 11 & 15 & 16 \end{bmatrix}$$

Тоді отримаємо наступні вектори:

$$\begin{aligned} V_1 &= [0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0\ 0\ 0\ 1] \\ V_2 &= [0\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1] \\ V_3 &= [0\ 0\ 0\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 1] \end{aligned}$$

Введемо поняття вектору груп  $W$  як вектора, елементами якого є кількості однакових елементів вектору  $V$ , що стоять поспіль. Вектор груп  $W$  є результатом процедури ущільнення маски і зберігається разом з іншими даними про зображення.

Для наведеного прикладу отримуємо такі вектори груп:

$$\begin{aligned} W_1 &= [5\ 5\ 5\ 1] \\ W_2 &= [2\ 1\ 2\ 2\ 2\ 1\ 3\ 1\ 1\ 1] \\ W_3 &= [4\ 2\ 1\ 2\ 1\ 1\ 4\ 1] \end{aligned}$$

З цього випливає, що для заданої матриці  $X$  найбільш прийнятним з точки зору довжини груп однакових елементів, що стоять поспіль, є спосіб перетворення, при якому матриця перетворюється на вектор шляхом послідовного запису її стовпців.

Узагальнимо цей спосіб на випадок довільної матриці.

Нехай дана матриця  $T$  виду:

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1m} \\ t_{21} & t_{22} & \dots & t_{2m} \\ \dots & \dots & \dots & \dots \\ t_{n1} & t_{n2} & \dots & t_{nm} \end{bmatrix}, \quad (3.144)$$

де  $t_{ij} \in \{0,1\}$ .

Отримаємо для кожного рядка матриці  $T$  вектор груп виду  $R^i = [r_1^i \ r_2^i \ \dots \ r_k^i]$ , де  $k$  – кількість груп однакових елементів, що стоять поспіль,  $i$  – номер рядка матриці  $T$ , згідно з такими умовами:



1. Вектор груп завжди складається з парної кількості елементів.
2. Елементи, що стоять на непарних позиціях вектору груп, означають кількість елементів у групах нулів, що стоять поспіль.
3. Якщо який-небудь стовпець або який-небудь рядок матриці починається з одиниці, то  $r_1^i = 0$ ,
4. Елементи, що стоять на парних позиціях вектору груп, означають кількість елементів у групах одиниць, що стоять поспіль.
5. Якщо який-небудь стовпець або який-небудь рядок матриці закінчується нулем, то  $r_k^i = 0$ .

Далі обчислюємо суму норм обчислюємо суму  $n$  векторів груп  $R^i$ :

$$S_R = ||R^1|| + ||R^2|| + \dots + ||R^n||. \quad (3.145)$$

Аналогічно отримуємо вектори груп  $C^j = [c_1^j \ c_2^j \ \dots \ c_p^j]$ , де  $p$  – кількість груп однакових елементів, що стоять поспіль,  $j$  – номер стовпця, для всіх стовпців матриці  $T$ . Після чого обчислюємо суму норм отриманих  $m$  векторів груп  $C^j$ :

$$S_C = ||C^1|| + ||C^2|| + \dots + ||C^m||. \quad (3.146)$$

З двох можливих способів перетворення матриці на вектор (шляхом послідовного запису стовпців та шляхом послідовного запису рядків) слід обрати перший, якщо  $S_R < S_C$ ; інакше слід обрати другий.

Тепер з множини векторів груп  $\{R^1, \dots, R^n\}$  або  $\{C^1, \dots, C^m\}$  (залежно від результату порівняння значень  $S_R$  та  $S_C$ ) необхідно отримати єдиний вектор груп  $W$  для всієї матриці  $T$ , що і буде результатом процедури ущільнення маски  $T$ .

Нехай в результаті порівняння значень  $S_R$  та  $S_C$  був обраний спосіб

перетворення матриці на вектор шляхом послідовного запису стовпців. Тоді необхідно визначити порядок запису стовпців, при якому довжина вектору  $W$  буде мінімальною, тобто для збереження маски потрібна мінімальна кількість байт. Для цього отримані  $m$  векторів груп  $C^j$ , що відповідають стовпцям матриці  $T$ , необхідно проаналізувати наступним чином:

1. Якщо  $c_p^i = 0$  та  $c_1^j \neq 0$ , де  $i$  та  $j$  – номери стовпців ( $i \neq j$ ), то ці стовпці можуть бути об'єднані у порядку  $\{C^j C^i\}$ , причому результуючий вектор груп буде мати вигляд:

$$W = [c_1^i \quad c_2^i \quad \dots \quad c_{p-1}^i \quad (c_p^i + c_1^j) \quad c_2^j \quad \dots \quad c_p^j]. \quad (3.147)$$

2. Якщо  $c_p^i \neq 0$  та  $c_1^j = 0$ , де  $i$  та  $j$  – номери стовпців ( $i \neq j$ ), то ці стовпці можуть бути об'єднані у порядку  $\{C^j C^i\}$ , причому результуючий вектор груп буде мати вигляд:

$$W = [c_1^j \quad c_2^j \quad \dots \quad c_{p-1}^j \quad (c_p^j + c_1^i) \quad c_2^i \quad \dots \quad c_p^i]. \quad (3.148)$$

Якщо існує декілька способів об'єднання стовпців матриці, то перевагу слід віддати тому способу, при якому довжина вектору  $W$  буде мінімальною. Найкращим способом є об'єднання стовпців в порядку їх входження в матрицю  $T$ . У цьому випадку порядок проходження стовпців можна не зберігати разом з ущільненою маскою.

Аналогічно виконуються дії над рядками матриці  $T$ . При цьому разом з даними про ущільнену маску необхідно зберігати ознаку того, чи були об'єднані рядки або стовпці матриці.

Запропонована процедура ущільнення маски є оборотною, оскільки:

- відомо, яким чином матриця  $T$  була перетворена в вектор: по рядках або по стовпцях, а також відомий порядок, в якому це перетворення

виконано,

- матриця  $T$  є бінарною,
- вектор довжин груп  $W$  має однозначно визначену структуру (парна кількість елементів; елементи, які стоять на непарних позиціях, означають кількість нулів, що стоять поспіль; елементи, які стоять на парних позиціях, означають кількість одиниць, що стоять поспіль).

Вихідними даними для процедури обробки даних є:

- зображення;
- маска або критерій її отримання;
- алгоритм ущільнення або критерій його вибору.

Процедура обробки даних включає такі основні дії:

1. Визначення маски.
2. Поділ масиву даних про зображення на два потоки: потік даних, що підлягають ущільненню, та потік даних без ущільнення.
3. Ущільнення потоку даних, що підлягають ущільненню, заданим або обраним за певними критеріями алгоритмом.
4. Ущільнення маски.
5. Запис даних у файл заданого формату.

Процес відновлення даних відбувається у зворотному порядку.

Дані про зображення, які необхідно зберігати для гарантованого відновлення зображення, включають в себе такі компоненти:

- послідовність ущільнених даних,
- послідовність неущільнених даних,
- ущільнена маска та порядок її відновлення.

Крім того, має бути збережений розмір зображення, кількість градацій кольору і ознака, що вказує на те, який алгоритм використовується для ущільнення даних, які підлягають ущільненню.

Обсяг ущільнених даних  $U_c$  та коефіцієнт ущільнення залежать від способу ущільнення. Обсяг неущільнених даних  $U_c = m \cdot n - N_c$ , де  $N_c$  –

кількість пікселів, дані про колір яких входять у послідовність, що підлягає ущільненню.

Формат збереження даних про зображення наведений у табл. 3.1.

Таблиця 3.1 – Структура файлу

Назва поля	Довжина (байти)	Зміщення (байти)	Опис
<i>Секція метаданих</i>			
Width	2	0	Ширина зображення
Height	2	2	Висота зображення
Colour Levels	2	4	Кількість градацій кольору
Planes	1	6	Кількість колірних площин
Compression Type	1	7	Тип алгоритму ущільнення
Compressed Data Offset	4	8	Зміщення ущільнених даних
Uncompressed Data Offset	4	12	Зміщення неущільнених даних
Mask Method	1	16	Ознака способу перетворення маски
Length Offset	4	17	Зміщення вектору довжин груп
Order Offset	4	21	Зміщення вектору порядку
<i>Секція даних</i>			
Compressed Data	$U_c$	25	Послідовність ущільнених даних
Uncompressed Data	$U_u$	$25 + U_c$	Послідовність неущільнених даних
Length	$k$	$25 + U_c + U_u$	Вектор довжин груп
Order	$q$	$25 + U_c + U_u + k$	Вектор порядку

Запропонований метод адаптивного ущільнення дозволяє забезпечити точність подання інформації про область підвищеного інтересу на зображенні при здійсненні його ущільнення для компактного зберігання графічних даних візуальної моделі цифрового двійника.

### 3.6. Метод конфіденційного зберігання даних цифрового двійника

Дані про досліджуваний об'єкт, які подаються за допомогою цифрового двійника, можуть мати конфіденційний характер. Якщо для зберігання таких даних використовуються такі сховища, як, наприклад, Azure Data Lake Storage [322], то конфіденційність даних гарантується механізмами захисту даних самого сховища. Проте, якщо обсяг даних є порівняно невеликим, то дані можуть зберігатись локально, але при цьому вони можуть потребувати захисту від несанкціонованого доступу. При цьому необов'язково всі дані цифрового двійника є однаково вразливими. Тому забезпечення конфіденційного зберігання для різних даних може досягатись різними способами, включаючи застосування алгоритмів легковагової криптографії (lightweight cryptography), які останнім часом почали використовуватись для захисту даних у застосунках IoT (Internet of things, Інтернет речей) [329–332], та алгоритмів стеганографії, які використовуються для захисту мультимедійних даних [8, 18–20, 23, 25, 29, 40, 50, 55–57, 61, 63, 333–335].

Сформулюємо метод конфіденційного зберігання даних цифрового двійника. Він складається з наступних етапів.

Першим етапом методу є оцінювання ступеню конфіденційності даних кожної модальності та визначення відповідної категорії даних.

Результатом цього етапу є призначення даним кожної модальності однієї з сімох категорій, що визначені, як наведено у табл. 3.2.

Оцінювання відкритості / конфіденційності даних відбувається експертним шляхом, виходячи з таких основних чинників:

- мета створення та галузь застосування цифрового двійника;
- структура моделі цифрового двійника;
- обсяг даних цифрового двійника;
- модальність (тип) даних.

Таблиця 3.2 – Категорії даних

Тип даних	Дані незначного обсягу	Дані великого обсягу
Дані, що характеризуються надлишковістю подання інформації		
Відкриті дані	Категорія 1	Категорія 2
Конфіденційні дані	Категорія 4	Категорія 5
Інші дані		
Відкриті дані	Категорія 3	
Конфіденційні дані	Категорія 6	Категорія 7

Визначення обсягу даних, який має вважатись великим, відбувається експертним шляхом, виходячи з можливостей обробки та зберігання даних.

Даними, що характеризуються надлишковістю подання інформації, є графічні дані (зображення, відео), акустичні дані, тактильні дані тощо.

На другому етапі відбувається визначення способу (процедури) конфіденційного подання для кожного виду даних цифрового двійника, виходячи з результату виконання першого етапу.

Можливі три способи конфіденційного подання мультимодальних даних цифрового двійника:

1. Шифрування даних на основі процедур легковагової криптографії [29]. Цей спосіб доцільно застосовувати для даних категорії 4.
2. Стеганографічне приховування конфіденційних даних [8, 16, 18–20, 23, 25, 40, 50]. Цей спосіб доцільно застосовувати для даних категорій 6. При цьому дані того ж самого цифрового двійника, які належать до категорій 1 та 2, доцільно використовувати як контейнер для приховування конфіденційних даних, оскільки це дозволить зменшити загальний обсяг даних, отриманих після виконання стеганографічного приховування.
3. Застосування специфічного захисту даних (зберігання даних у сховищі даних із застосуванням відповідних засобів захисту). Цей

спосіб доцільно застосовувати для даних категорій 5 та 7.

Дані категорії 3 не потребують конфіденційного зберігання та не можуть використовуватись як контейнер для конфіденційного зберігання інших даних.

Третім етапом методу є застосування обраного способу для конфіденційного подання даних кожної модальності.

Розглянемо докладно процедури, що реалізують перший та другий способи конфіденційного подання мультимодальних даних цифрового двійника, а саме, процедуру шифрування даних з використанням палітри значень та процедуру стеганографічного приховування даних з використанням булевих функцій.

### **3.6.1. Процедура шифрування даних з використанням палітри значень**

Процедура шифрування даних з використанням палітри значень [29] ґрунтується на шифруванні значень (елементів послідовності конфіденційних даних) шляхом їх заміни адресами таких самих або близьких їм значень у файлі-ключі, який називатимемо *палітрою значень*, де представлені всі можливі значення, які присутні у послідовності конфіденційних даних, що шифрується.

В ролі палітри значень виступає файл, який в процесі шифрування не зазнає жодних змін. Оскільки цей файл не має певної специфічної структури та не містить жодних вбудованих повідомлень, він є відкритим та може бути переданий користувачу конфіденційних даних будь-яким способом (електронна пошта, файлове сховище, соціальні мережі тощо), але приховуючи від сторонніх осіб роль цього файлу для процесу обробки конфіденційних даних. Для шифрування та дешифрування

використовується один і той самий файл-ключ, тобто криптосистема є симетричною.

Вимогою до файлу даних для використання в ролі палітри значень є наявність серед даних, які у ньому містяться, повного діапазону всіх можливих значень, що присутні у послідовності конфіденційних даних (звідси взялась назва – «палітра значень», що набуває буквального сенсу при застосуванні цього підходу для захисту зображень [29]). Після перевірки виконання цієї вимоги відбувається процес створення словника адрес унікальних значень палітри.

Оскільки конфіденційні дані, що передбачається оброблювати за допомогою цієї процедури, належать до категорії 4 (табл. 3.2), тобто є такими, що характеризуються надлишковістю подання інформації, то можливе використання у процедурі обробки конфіденційних даних значень з файлу-ключа, що відхиляються від значення елементу послідовності конфіденційних даних на величину припустимої похибки. Наприклад, якщо конфіденційними даними є зображення, то можлива похибка у рівні яскравості колірної компоненти відтінку кольору пікселя на  $1 \leq \delta \leq 4$ , оскільки людська зорова система не відрізняє відтінки кольорів, близькі за значенням рівня яскравості.

Нехай потрібно обробити  $j$ -ту послідовність конфіденційних даних цифрового двійника  $\langle d_i^j \rangle_{i=1}^{n_j}$ ,  $d_i^j \in M_j$ , використовуючи як палітру файл  $F = \langle f_l \rangle_{l=1}^R$ ,  $f_l \in M_j$ . При цьому має виконуватись наступна *вимога повноти палітри*:  $\forall i, i = [1 \dots n_j], \exists f_l = d_i^j \pm \delta, l = [1 \dots R]$ , де  $\delta \geq 0$  – це припустима похибка визначення значення  $d_i^j$ . Тоді визначимо словник як:

$$\Omega = [\langle f_g, u_g \rangle]_{g=1}^Q, \quad (3.149)$$



де  $u_g = [\alpha_q^g]_{q=1}^{h_g}$  – список адрес  $\alpha_q^g$ , за якими значення  $f_g$  зустрічається у файлі  $F$ ;  $h_g$  – кількість разів, скільки  $f_g$  зустрічається у файлі  $F$ ;  $Q$  – кількість унікальних значень у послідовності  $\langle d_i^j \rangle_{i=1}^{n_j}$ .

У процесі шифрування замість кожного значення даних  $d_i^j$ , у вихідний файл  $W$  записується значення адреси  $\alpha_q^G$ , за якою у файлі  $F$  знаходиться значення  $f_l = d_i^j \pm \delta$ , причому  $G$ , яке дозволяє отримати адресу  $\alpha_q^G$  зі списку адрес  $u_g$ , є деяким випадковим значенням, таким, що  $G \in [1 \dots h_g]$ . Тоді формула шифрування значення  $d_i^j$  має вигляд:

$$w_i = \Omega(f_i, u_i(G)). \quad (3.150)$$

Для відновлення зашифрованих даних з файлу  $W$  необхідно мати файл-ключ  $F$ . В процесі відновлення даних відбувається послідовне зчитування значень з файлу  $W$ , який містить адреси  $\alpha_q$ , та відновлення значень конфіденційних даних. Формула відновлення має вигляд:

$$\hat{d}_i^j = F(\alpha_q), \quad (3.151)$$

де  $\hat{d}_i^j$  –  $i$ -те значення  $j$ -ї послідовності конфіденційних даних цифрового двійника, відновлене з припустимою похибкою  $\delta$ .

Якщо  $\delta = 0$ , то отримаємо вхідну послідовність даних  $\langle d_i^j \rangle_{i=1}^{n_j}$ , інакше конфіденційні дані відновлюються із заданою похибкою.

Особливістю цієї процедури те, що у ролі ключа процедури шифрування використовується звичайний файл (проте такий, що задовольняє вимогу повноти палітри). Така особливість ускладнює несанкціонований підбір ключа та підвищує надійність захисту.

### 3.6.2. Процедура стеганографічного приховування даних з використанням булевих функцій

Стеганографічний спосіб приховування конфіденційних даних передбачає, що ці дані вбудовуються у *контейнер*, який являє собою файл даних, що характеризуються надлишковістю подання інформації. До таких даних відноситься більшість мультимедійних даних: графічні дані (зображення, відео), акустичні дані, тактильні дані. Найкращим типом даних для використання як контейнер є графічні дані, оскільки вони характеризуються значною надлишковістю та звичайно мають достатній для приховування конфіденційних даних об'єм [25, 40]. Контейнер має бути обраний з урахуванням можливості вбудовування до нього заданої послідовності конфіденційних даних [18–20].

Оскільки звичайне стеганографічне приховування само по собі не гарантує надійного захисту конфіденційних даних, звичайною практикою [8, 18–20, 23, 25, 29, 40, 50, 55–57, 61, 63, 333–335] є кодування конфіденційних даних перед їх вбудовуванням у контейнер.

У процедурі, що пропонується, вбудовування конфіденційних даних відбувається шляхом їх по-бітного запису до найменш значущих бітів (LSB) значень, що зберігаються у контейнері. Перед вбудовуванням до кожного біту значення послідовності конфіденційних даних та одного чи декількох найбільш значущих бітів (MSB) значення з контейнеру застосовується деяка булева функція.

Якщо використовується лише один MSB, то булева функція має два аргументи: біт  $S$  таємних даних і біт  $C$ , що є MSB. В результаті отримуємо біт  $E$ , який буде використовуватися для заміни певного біта LSB у послідовності даних контейнера. У цьому випадку можливе використання виключної диз'юнкції або еквіваленції. Решту булевих функцій неможна використовувати для кодування даних через неможливість подальшого

відновлення даних [40]. Кількість булевих функцій, придатних для кодування даних, залежить від кількості  $m$  MSB:

$$n = 2^{2^m} - 2. \quad (3.152)$$

Значення  $m$  є частиною складеного ключа, що використовується для розшифровки конфіденційних даних.

Принцип модифікації LSB наведено на рис. 3.17.

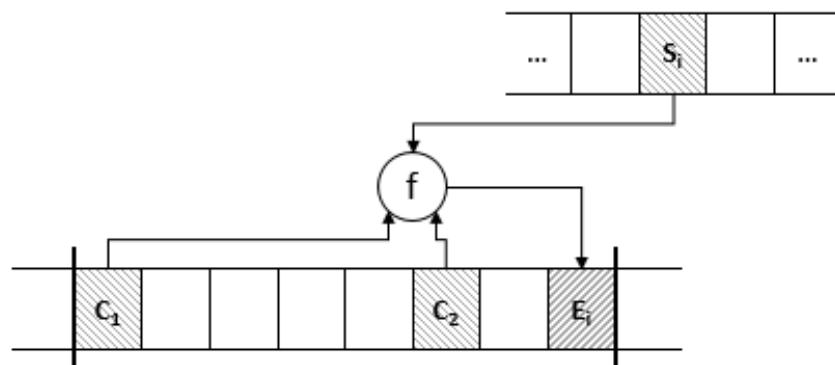


Рис. 3.17. Принцип модифікації LSB

Результатом застосування обраної булевої функції є біт  $E_i$ , який використовуються для заміни LSB у черговому значенні з контейнера.

Процедура стеганографічного приховування складається з таких кроків:

1. Перетворення конфіденційних даних на послідовність бітів.
2. Вбудовування даних у контейнер за принципом, показаним на рис. 3.17.
3. Формування складеного ключа з наступних компонентів:
  - Кількість елементів конфіденційних даних;
  - Кількість MSB, що використовуються в процедурі кодування даних, та їхні порядкові номери у байті;
  - Ідентифікатор булевої функції, що використовується для кодування.

Оскільки ключ є секретним, то він має передаватись користувачу конфіденційних даних відповідно до загальноприйнятої практики маніпулювання секретними ключами.

Процедура відновлення конфіденційних даних протилежна описаній процедурі приховування конфіденційних даних.

Ймовірність несанкціонованого доступу до конфіденційних даних шляхом відгадування місця розташування бітової послідовності цих даних у послідовності бітів контейнера можна оцінити наступним чином:

$$p_{loc} = \frac{q! \cdot (n - q)! \cdot m! \cdot (8 - k - m)!}{n! \cdot (8 - k)!} \rightarrow 0 \quad (5.24)$$

де  $k$  – кількість використаних LSB;

$m$  – кількість використаних MSB;

$n$  – кількість бітів контейнеру;

$q$  – кількість бітів конфіденційних даних.

Таким чином, запропонована процедура забезпечує належний захист конфіденційних даних цифрового двійника.

### 3.7. Висновки до розділу 3

У цьому розділі розроблено моделі подання та методи обробки темпоральних мультимодальних даних цифрового двійника досліджуваного об'єкта. Зокрема, отримано такі результати.

1. Розроблено концепцію мультиобразу для формальної специфікації моделей цифрового двійника. Мультиобраз є математичною основою моделей цифрового двійника для подання темпоральних мультимодальних даних, які характеризують та визначають досліджуваний об'єкт. Показано та обґрунтовано, що до мультиобразу можуть бути застосовані всі операції

та відношення, визначені в алгебраїчній системі агрегатів, що дозволяє застосовувати апарат алгебраїчної системи агрегатів для опису та обробки темпоральних мультимодальних даних цифрового двійника. Продemonстровано принцип застосування концепції мультиобразу для формальної специфікації мультимедійної візуальної моделі цифрового двійника.

2. Розроблено метод створення мультиобразу досліджуваного об'єкта, вхідними даними якого є окремі набори темпоральних даних. Метод складається з сімох етапів, які передбачають формування структури даних мультиобразу досліджуваного об'єкта, декомпозицію специфікації мультиобразу на набір специфікацій часткових мультиобразів, отримання та підготовку окремих наборів даних, об'єднання часткових мультиобразів у єдиний мультиобраз, сортування мультиобразу та проріджування відсортованого мультиобразу за кортежем часових значень. Результатом методу є мультиобраз досліджуваного об'єкта, що містить синхронізовані та агреговані послідовності темпоральних мультимодальних даних, які характеризують цей досліджуваний об'єкт.

3. Розроблено метод синхронізації мультиобразів, який ґрунтується на застосуванні інтервальних відношень та виконанні правил синхронізації. Правила синхронізації включають універсальне правило синхронізації, базові правила синхронізації, правила нечіткої синхронізації. Метод передбачає формування шаблону синхронізації, що складається з декількох правил синхронізації, які відповідають виду періодичності у часових кортежах, що синхронізуються. Вхідними даними для методу є мультиобрази, припустима похибка визначення часу та тип послідовності часових значень. Результатом методу є синхронізовані мультиобрази, готові до подальшої обробки відповідно до мети дослідження об'єкта.

4. Розроблено три моделі цифрового двійника: муксельну модель, часову зв'язну модель та багаторівневу онтологічну модель. Ці моделі

дозволяють представляти цифровий двійник досліджуваного об'єкта на різних рівнях абстракції: як суцільний об'єкт (муксельна модель), як об'єкт, що визначається набором станів, динамічно змінюваних з плином часу (часова зв'язна модель); як об'єкт, що є композицією своїх складових, кожна з яких в свою чергу може розглядатись як окремий об'єкт, утворюючи ієрархію об'єктів (багаторівнева онтологічна модель). Для кожної моделі визначено структуру та запропоновано формати зберігання даних. Запропоновані моделі можуть комбінуватись для отримання найбільш інформативного способу опису фізичного двійника.

5. Розроблено метод адаптивного ущільнення графічних даних цифрового двійника, який ґрунтується на застосуванні морфологічних операцій до графічних даних. Метод передбачає визначення у зображенні однієї чи декількох областей інтересу з метою уникнення втрати якості візуалізації важливих елементів, присутніх у зображенні досліджуваного об'єкта, яке відбувається внаслідок ущільнення графічних даних. Метод призначений для компактного зберігання графічних даних візуальної моделі цифрового двійника із забезпеченням подальшої якісної візуалізації.

6. Розроблено метод конфіденційного зберігання даних цифрового двійника, особливістю якого є застосування процедури забезпечення конфіденційності даних залежно від категорії даних. Запропоновано процедуру шифрування даних з використанням палітри значень та процедуру стеганографічного приховування даних з використанням булевих функцій. Процедура шифрування даних з використанням палітри значень ґрунтується на шифруванні елементів послідовності конфіденційних даних шляхом їх заміни адресами значень у файлі-ключі (палітрі значень). Процедура стеганографічного приховування даних з використанням булевих функцій ґрунтується на вбудовуванні у файл-контейнер результату застосування деякої обраної булевої функції до бітів конфіденційних даних та бітів даних контейнеру.

## **РОЗДІЛ 4. ПАРАДИГМА ПРОГРАМУВАННЯ МУЛЬТИБРАЗІВ ДЛЯ ОБРОБКИ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВИХ ДВІЙНИКІВ**

Одним з чинників, які впливають на ефективність обробки даних у комп'ютерній системі, є оптимальність коду програмного забезпечення, що залежить від стилю програмування та моделі обчислень, яка визначається парадигмою програмування.

*Парадигма програмування* – це сукупність принципів, методів і понять, що визначають спосіб конструювання програм. Парадигма програмування визначає підхід до програмування на основі математичної теорії або узгодженого набору принципів. Кожна парадигма підтримує набір понять, що робить її найкращою для вирішення задач певного класу [336]. Так, об'єктно-орієнтоване програмування спрямоване на вирішення задач, що характеризуються великою кількістю пов'язаних абстракцій даних, організованих в ієрархії. Логічне програмування призначене для вирішення задач за правилами логічного виведення. Програмування систем реального часу орієнтоване на вирішення задач, які вимагають реакції у реальному часі з чітко визначеними вимогами до часу обробки даних.

Як показав аналіз існуючих парадигм програмування, результати якого представлено у розділі 1, жодна з існуючих парадигм програмування не задовольняє у повній мірі вимоги щодо обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів.

Розробимо та реалізуємо парадигму програмування, яка спрямована на створення програмного забезпечення для обробки темпоральних мультимодальних даних, зокрема, даних цифрових двійників досліджуваних об'єктів. Оскільки ця парадигма ґрунтується на концепції мультимобразу, називатимемо її *парадигмою програмування мультимобразів*.

#### 4.1. Парадигма програмування мультиобразів

Парадигма програмування мультиобразів орієнтована на вирішення задач обробки темпоральних мультимодальних даних, для яких характерні такі основні особливості:

1. При отриманні (визначені, вимірюванні, генерації) мультимодальних даних фіксується момент часу, коли це отримання відбулось.
2. Дані різних модальностей, що визначають певний досліджуваний об'єкт, розглядаються та оброблюються у сукупності з даними інших модальностей, що визначають той самий об'єкт.
3. Дані отримуються з різномірних джерел та у довільних форматах.

Парадигма програмування мультиобразів, як методологія створення програмного забезпечення, полягає в тому, що розробник має враховувати взаємозв'язок даних з точки зору *інтервалу часу* їх існування (отримання, визначення, генерації, вимірювання тощо) та *модальності* даних, тобто оперувати такою сутністю як *мультиобраз* об'єкта.

Структура даних для опису мультиобразу має бути визначена у основній частині програмного коду, з якої починається вирішення задачі обробки даних. Логіка вирішення цієї задачі має ґрунтуватись на операціях та відношеннях, що визначені для мультиобразу як математичного об'єкта, тобто на апараті алгебраїчної системи агрегатів.

Програмний код має бути побудований таким чином, щоб розробник міг максимально абстрагуватись від того, що є джерелом даних, у якому форматі ці дані поступають для оброблення та що буде виступати у ролі приймача даних при їх відтворенні. Задачі перетворення даних між форматами, підготовки до відтворення за допомогою спеціалізованого обладнання вирішуються за допомогою спеціалізованих бібліотек, як мають поставлятися розробниками програмного забезпечення цього спеціалізованого обладнання.



Характерними особливостями парадигми програмування мультимедіа є наступні:

1. Основною сутністю, на якій ґрунтується подання та обробка даних у програмі, є мультимедіа.
2. Синхронізація та агрегація даних різних модальностей є основою моделі обчислення для обробки даних мультимедіа.
3. Процес обробки ґрунтується на врахуванні властивості темпоральності даних.
4. Обробка мультимедіа здійснюється з використанням операцій та відношень, визначених в алгебраїчній системі агрегатів.

Крім того, абстрагування понять джерела та приймача даних виконується за рахунок використання спеціалізованих бібліотек, які дозволяють виконувати перетворення даних з однієї форми подання у іншу форму (формат) без необхідності для програміста у розробленні власного програмного коду для підготовки даних.

Розглянемо моделі обчислень [2], які дозволяють реалізувати цю парадигму програмування.

#### **4.2. Моделі обчислень для реалізації парадигми програмування мультимедіа**

*Модель обчислень* є формальною системою, яка визначає, як здійснюються обчислення [337]. Модель обчислень конкретизує більш узагальнене поняття парадигми програмування. Визначимо базову модель обчислень для обробки довільних темпоральних мультимодальних даних та модель обчислень для обробки даних досліджуваного об'єкта за технологією цифрових двійників.

#### 4.2.1. Базова модель обчислень

Базова модель обчислень [2] ґрунтується на понятті мультиобразу як абстрактної структури даних. Як визначено у розділі 2, мультиобраз – це агрегат алгебраїчної системи агрегатів, першим кортежем якого є непустий кортеж часових даних.

В базовій моделі обчислень (рис. 4.1) мультиобраз є ключовою абстракцією, яка визначається на двох рівнях: на рівні обробки окремих вхідних темпоральних даних (часткові мультиобрази) та на рівні формування єдиної синхронізованої та агрегованої сукупності темпоральних мультимодальних даних (мультиобраз об'єкта), що підлягає подальшій обробці.

На першому рівні мультиобраз визначається як агрегат, що складається з двох кортежів: кортежу часових значень (часових міток) та кортежу даних певної модальності. На цьому рівні відбувається первинна синхронізація даних – синхронізація часових міток та даних модальності, для якої вони визначені. Таким чином, структура мультиобразу першого рівня є типовою, проте другий кортеж кожного з цих мультиобразів належить до певного специфічного типу даних.

На другому рівні мультиобраз визначає спосіб агрегації всіх мультимодальних даних у вигляді однієї структури даних. Мультиобраз другого рівня відповідає деякій моделі подання даних, наприклад, однієї із запропонованих у розділі 3.

Отже, схема базової моделі обчислень включає дворівневу ієрархічну процедуру синхронізації та агрегації. На першому рівні синхронізація та агрегація відбувається для пари кортежів «час» – «модальність», а на другому рівні синхронізуються дані всіх модальностей та відбувається їх агрегація у єдину впорядковану сукупність даних, готову для подальшої обробки. Ця обробка є третім рівнем схеми базової моделі обчислень.

Така схема обчислень дозволяє спростити синхронізацію та агрегацію різнорідних даних, що надходять з різних джерел. Синхронізація відбувається згідно з методом створення мультиобразу та методом синхронізації мультиобразів, розроблених у розділі 3.

Подальшу обробку даних отриманих мультиобразів передбачається виконувати із застосуванням апарату алгебраїчної системи агрегатів, а саме, за допомогою логічних операцій, операцій впорядкування та відношень агрегатів та їх компонентів (кортежів та окремих елементів), які визначено у розділі 2.

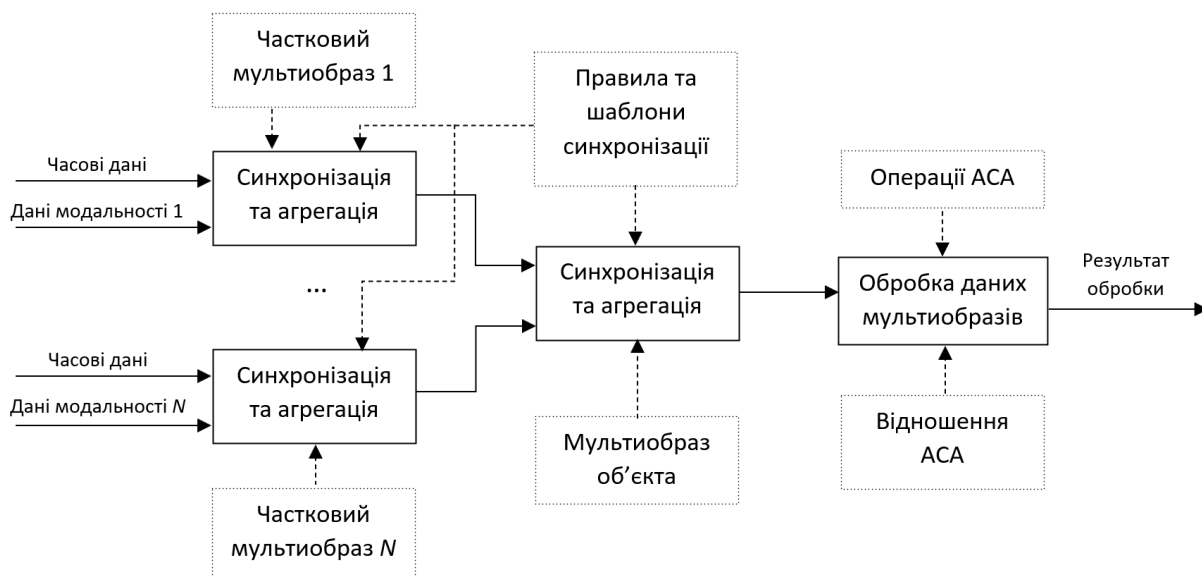


Рис. 4.1. Схема базової моделі обчислень

Результат обробки вхідних темпоральних мультимодальних даних може являти собою мультиобраз, окремі елементи темпоральних мультимодальних даних, команди для взаємодії з об'єктом, чисельні або текстові аналітичні дані тощо.

#### 4.2.2. Модель обчислень для технології цифрових двійників

Обробка мультимодальних даних цифрових двійників [2] має свої особливості, пов'язані з тим, що цифровий двійник складається з двох моделей: поведінкової моделі та візуальної моделі. Відповідно дані кожної з цих моделей мають різну природу та різне джерело. Тому доцільним є введення ще одного, середнього, рівня ієрархічної процедури синхронізації та агрегації, на якому відбувається синхронізація та агрегація мультимодальних даних у межах однієї моделі: поведінкової або візуальної. Слід підкреслити, що візуальна модель подає не лише зовнішній вигляд досліджуваного об'єкта, а включає всі мультимедійні модальності для надання комплексного подання досліджуваного об'єкта.

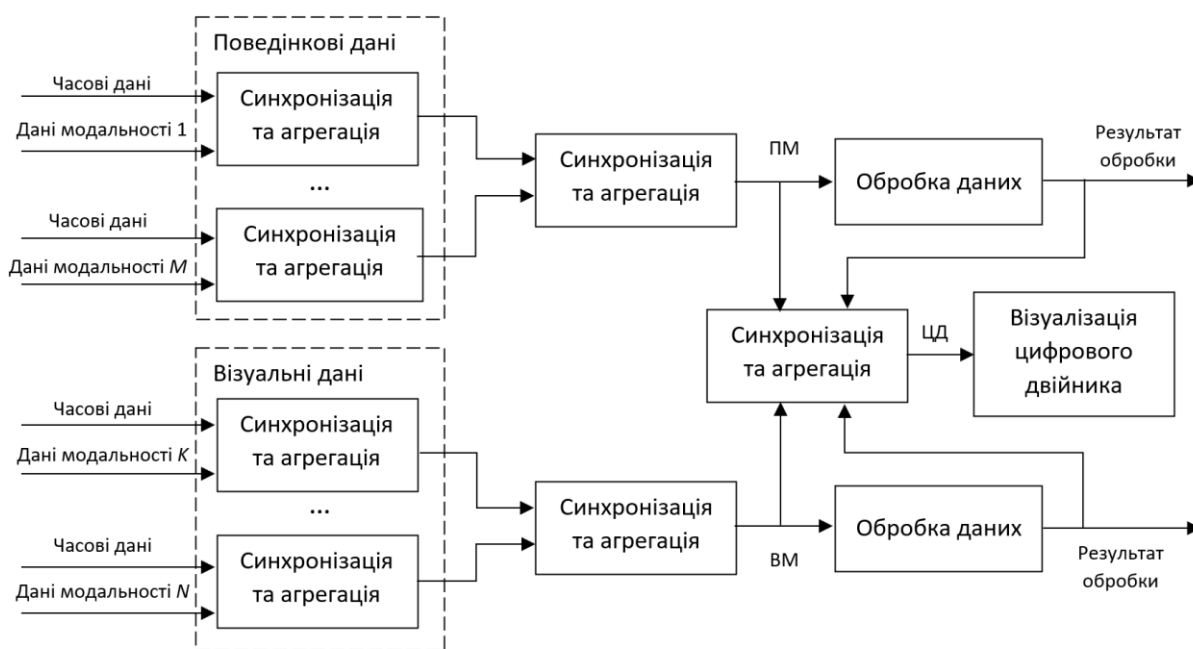


Рис. 4.2. Схема моделі обчислень для цифрових двійників

Згідно зі схемою моделі обчислень для цифрових двійників (рис. 4.2) на першому етапі відбувається синхронізація та агрегація даних кожної модальності окремо. Зміст цього етапу полягає у тому, щоб синхронізувати

дані однієї і тієї ж самої модальності, які можуть поступати з декількох джерел. Наприклад, у програмно-апаратних системах, що реалізують технологію IoT у сільському господарстві, дані про вологість ґрунту можуть поступати з декількох давачів, розташованих у різних місцях сільськогосподарської земельної ділянки [300].

На другому етапі відбувається формування поведінкової моделі та візуальної моделі досліджуваного об'єкта. Для кожної з цих моделей синхронізується та агрегується свій набір мультимодальних даних.

На наступному, третьому, етапі відбувається синхронізація та агрегація даних поведінкової моделі та даних візуальної моделі, в результаті чого створюється складний віртуальний об'єкт, який є цифровим двійником досліджуваного об'єкта. Створення цифрового двійника та його відтворення (візуалізація) є одним з результатів обробки вхідних мультимодальних даних. Водночас, дані кожної з моделей (поведінкової та візуальної) оброблюються для вирішення певних задач дослідження. Такими задачами можуть бути прогнозування поведінки досліджуваного об'єкта, моделювання можливих станів досліджуваного об'єкта при зміні параметрів оточуючого середовища тощо.

Отже, модель обчислень для технології цифрових двійників включає в себе три етапи синхронізації та агрегації, етап обробки даних поведінкової та візуальної моделі відповідно до конкретної задачі дослідження об'єкта та етап відтворення цифрового двійника.

Таким чином, модель обчислень для технології цифрових двійників ґрунтується на базовій моделі обчислень темпоральних мультимодальних даних, проте відрізняється від неї більшою кількістю етапів обробки даних, що дозволяє підвищити ефективність обробки даних та спростити розроблення програмного забезпечення систем, що ґрунтуються на застосуванні технології цифрових двійників.

### 4.3. Застосування парадигми програмування мультиобразів

Запропонована парадигма програмування може бути реалізована як за допомогою універсальних мов програмування, так і за допомогою спеціалізованої мови програмування ASAMPL, розроблення якої виконане в цьому дослідженні.

При розробленні програмного коду відповідно до парадигми програмування мультиобразів потрібно визначити мультиобрази всіх рівнів та процедури синхронізації та агрегації темпоральних мультимодальних даних.

Типова структура, що визначає мультиобраз на першому етапі синхронізації та агрегації, у випадку реалізації запропонованої парадигми за допомогою універсальної мови програмування, наприклад, C#, має вигляд, наведений у лістингу 4.1.

Лістинг 4.1 – Визначення мультиобразу для першого етапу синхронізації та агрегації

```
class MultiImage<T>
{   public DateTime Time { get; set; }
    public T DataModality { get; set; }
    public MultiImage(DateTime time, T
        dataModality)
    {   Time = time;
        DataModality = dataModality; } }
```

де клас *MultiImage<T>* визначає *i*-й мультиобраз; *Time* визначає часові дані; *DataModality* визначає дані *i*-ї модальності; *T* визначає конкретний тип даних відповідно до модальності, дані якої зберігаються у цьому класі.

Тип *T* може визначатись структурою, що зберігає кортеж даних, елементи якого у свою чергу визначаються певними типами даних або

структурами. Наприклад, елементи кортежу даних воксельної моделі визначаються структурою, що наведена у лістингу 4.2.

Лістинг 4.2 – Визначення елемента даних воксельної моделі

```
struct Voxel
{
    public int x;
    public int y;
    public int z;
    public int color;
    public Voxel(int x, int y, int z, int color)
    {
        this.x = x;
        this.y = y;
        this.z = z;
        this.color = color; } }
```

Тоді можемо створити мультиобраз *MultiImage<Voxel>*, який скрадатиметься з мультимодальних даних типу *Voxel*, що описаний структурою у лістингу 4.2, та часових даних. У лістингу 4.3 наведено приклад ініціалізації такого мультиобразу.

Лістинг 4.3 – Приклад створення екземпляру мультиобразу

```
Voxel currentVoxel = new Voxel(14, 8, 5, 201);
DateTime currentTime = DateTime.Now;
MultiImage<Voxel> multiImage1 = new
MultiImage<Voxel>(currentTime, currentVoxel);
```

Структура, що визначає мультиобраз на другому етапі базової моделі обчислень та включає, крім часових даних, дані  $N$  модальностей, має вигляд, наведений у лістингу 4.4.

Лістинг 4.4 – Визначення мультиобrazу для другого етапу синхронізації та агрегації

```
class MultiImage<DM1, DM2, ..., DMN>
{   public DateTime Time { get; set; }
    public DM1 DataModality1 { get; set; }
    public DM2 DataModality2 { get; set; }
    .....
    public DMN DataModalityN { get; set; }
    public MultiImage(DateTime time, DM1
dataModality1, DM2 dataModality2, ... DMN
dataModalityN)
    {   Time = time;
        DataModality1 = dataModality1;
        DataModality2 = dataModality2;
        .....
        DataModalityN = dataModalityN; } }
```

Таким чином, дані всіх модальностей в результаті синхронізації другого етапу приводяться до однієї часової шкали.

Процедура синхронізації даних реалізується як метод *Synchronize*, що ґрунтується на певних правилах та шаблонах синхронізації, зокрема, на правилах чіткої та нечіткої синхронізації, запропонованих у розділі 3.

Узагальнений вигляд методу *Synchronize\_i* першого етапу синхронізації та агрегації показаний у лістингу 4.5.

Лістинг 4.5 – Визначення методу *Synchronize\_i* першого етапу синхронізації та агрегації

```
public MultiImage<T> Synchronize(DateTime time,
Tuple<T> dataModality)
{   MultiImage<T> multiImage = new
    MultiImage<T>();
    /* synchronization rule realization */
    return multiImage; }
```



Узагальнений вигляд методу *Synchronize* другого етапу синхронізації та агрегації базової моделі обчислень показаний у лістингу 4.6.

Лістинг 4.6 – Визначення методу *Synchronize* другого етапу синхронізації та агрегації

```
public MultiImage<DM1,DM2, ..., DMN>
Synchronize(DateTime time, Tuple<DM1>
dataModality1, Tuple<DM2> dataModality2, ...,
Tuple<DMN> dataModalityN)
{   MultiImage< DM1,DM2, ..., DMN > multiImage = new
    MultiImage< DM1,DM2, ..., DMN >();
    /* synchronization rule realization */
    return multiImage; }
```

Окрема процедура агрегації може бути не потрібна, оскільки фактично агрегація відбувається одночасно із синхронізацією даних. Обробка синхронізованих та агрегованих мультимодальних даних виконується за допомогою набору класів та методів, які реалізують процедуру вирішення основної задачі обробки темпоральних мультимодальних даних досліджуваного об'єкта.

Таким чином, модель обчислень парадигми програмування для обробки структур мультимодальних даних може бути реалізована за допомогою універсальної мови програмування.

Як зазначалось вище, запропонована парадигма програмування також може бути реалізована за допомогою спеціалізованої мови програмування.

#### **4.4. Спеціалізована мова програмування ASAMPL для обробки темпоральних мультимодальних даних**

Мова програмування ASAMPL [4, 6, 21] є спеціалізованою мовою програмування, призначеною для обробки мультимедійних та

мультимедійних даних. Ця мова програмування отримала свою назву від словосполучень «Algebraic System of Aggregates» (алгебраїчна система агрегатів) та «Multimedia data Processing Language» (мова обробки мультимедійних даних).

Метою розроблення мови ASAMPL є спрощення обробки мультимодальних, зокрема, мультимедійних даних, основною особливістю яких є те, що вони є темпоральними. Крім того, обробка таких даних має враховувати модальність, їх сумісність даних різних модальностей та можливу взаємодію між ними. Тому основним принципом організації програми мовою ASAMPL є те, що виконання програми ґрунтується на двох сутностях: часі та модальності даних, що відповідає принципам парадигми програмування мультимедіа.

Ключовою концепцією в мові ASAMPL є концепція мультимедіа об'єкта. Для подання мультимедіа об'єкта в цій мові програмування передбачені спеціальні структури даних – кортежі та агрегати. Обробка агрегатів виконується відповідно до правил, призначених в алгебраїчній системі агрегатів (АСА).

Основні особливості мови ASAMPL, які конкретизують принципи парадигми програмування для обробки структур мультимодальних даних:

1. Зв'язок процесу обробки даних з часовою шкалою.
2. Синхронізація даних різних модальностей.
3. Подання мультимодальних даних за допомогою агрегатів та кортежів.
4. Орієнтація на одночасне використання різних джерел мультимодальних даних та різних форматів.
5. Використання апарату АСА для обробки мультимедіа об'єктів та їхніх компонентів.

Розглянемо синтаксис, семантику та прагматику мови ASAMPL.

#### 4.4.1. Синтаксис мови ASAMPL

Програма мовою ASAMPL складається з блоків, які починаються з ключових слів (табл. 4.1).

Таблиця 4.1 – Ключові слова мови ASAMPL

Ключове слово	Значення
AGGREGATES	Початок блоку, який містить опис агрегатів, що утворені з визначених кортежів
HANDLERS	Початок блоку, який містить опис вбудованих та зовнішніх модулів для перетворення файлів з даними певної модальності
LIBRARIES	Початок блоку, який містить опис вбудованих та зовнішніх бібліотек
ACTIONS	Початок блоку, який визначає дії з оброблення даних
PROGRAM	Початок програмного коду
RENDERERS	Початок блоку, який містить опис вбудованих та зовнішніх модулів для рендерінгу даних певної модальності
SETS	Початок блоку, який містить опис множин
SOURCES	Початок блоку, який містить опис джерел даних, тобто ресурсів, де знаходяться мультимодальні дані
TUPLES	Початок блоку, який містить опис кортежів та їх належність до визначених множин
ELEMENTS	Початок блоку, який містить опис змінних та їх належність до визначених множин

Програма мовою ASAMPL має структуру, наведено у лістингу 4.7.

Лістинг 4.7 – Приклад програмного коду мовою ASAMPL для обробки та формування мультимедійного контенту

```
PROGRAM name {
LIBRARIES {
    library_1 is path_to_library_1;
    library_2 is path_to_library_2;
    ..... }
HANDLERS {
    handler_1 is path_to_handler_1;
    handler_2 is path_to_handler_2;
    ..... }
RENDERERS {
    renderer_1 is path_to_renderer_1;
    renderer_2 is path_to_renderer_2;
    ..... }
SOURCES {
    source_1 is path_to_source_1;
    source_2 is path_to_source_2;
    ..... }
SETS {
    set_1 is type_1;
    set_2 is type_2;
    ..... }
ELEMENTS {
    element_1 is set;
    element_2 = value;
    ..... }
TUPLES {
    tuple_1 = set_1;
    tuple_2 = set_2;
    ..... }
AGGREGATES {
    aggregate_1 = [tuple_1, ..., tuple_M];
    aggregate_2 = [tuple_K, ..., tuple_N];
    ..... }
ACTIONS {
    list_of_statements    } }
```

Для опису граматики мови програмування ASAMPL будемо використовувати розширену форму Бекуса-Наура (EBNF) [299]. Тоді програма мовою ASAMPL має синтаксис, що визначається синтаксичним правилом [SR1].

*програма* = PROGRAM , ідентифікатор , "{" , [SR1]  
LIBRARIES , "{" , { ідентифікатор , ( IS | "=" ) ,  
посилання , ";" } [ "/" , коментар ] "}" , ,  
HANDLERS , "{" , { ідентифікатор , ( IS | "="  
) , посилання , ";" } [ "/" , коментар ] "}" , ,  
RENDERERS , "{" , { ідентифікатор , ( IS | "="  
) , посилання , ";" } [ "/" , коментар ] "}" ,  
SOURCES , "{" , { ідентифікатор , ( IS | "=" ) ,  
посилання , ";" } [ "/" , коментар ] "}" ,  
SETS , "{" , { ідентифікатор , ( IS | "=" ) , тип ,  
";" } [ "/" , коментар ] "}" ,  
ELEMENTS , "{" , { ідентифікатор , ( IS | "=" ) ,  
( тип\_множини | значення ) , ";" } [ "/" ,  
коментар ] "}" , TUPLES , "{" , {  
ідентифікатор , ( IS | "=" ) , тип\_множини , ";"  
} [ "/" , коментар ] "}" , AGGREGATES "{" , {  
ідентифікатор , ( IS | "=" ) , "[" , ( ідентифікатор  
| кортеж\_значень , ";" } [ "/" , коментар ] "}" ,  
ACTIONS , "{" , { оператор , ";" } [ "/" ,  
коментар ] "}" , [ "/" , коментар ] , "}" ;

У мові ASAMPL визначено наступні оператори:

1. TIMELINE
2. SEQUENCE

3. IF THEN
4. CASE OF
5. SUBSTITUTE FOR WHEN
6. DOWNLOAD FROM
7. UPLOAD TO
8. IS
9. RENDER WITH.

Часовий оператор TIMELINE є специфічним оператором, який призначений для одночасного (паралельного) виконання дій протягом заданого періоду часу. Будемо розрізняти наступні види цього оператора:

- TIMELINE *значення1* : *крок* : *значення2* {*список дій з обробки даних, що виконуються одночасно*}
- TIMELINE AS *кортеж часових значень* {*список дій з обробки даних, що виконуються одночасно*}
- TIMELINE UNTIL *умова* {*список дій з обробки даних, що виконуються одночасно*}.

Часовий оператор визначається синтаксичним правилом [SR2].

*часовий\_оператор* = TIMELINE , (*ідентифікатор* [SR2]  
 | *значення\_часу* ) , ":", ( *ідентифікатор* |  
*значення\_часу* ) , ":" , ( *ідентифікатор* |  
*значення\_часу* ) , "{" , { *оператор* } , "  
 | TIMELINE , AS , *кортеж\_значень\_часу* , "{" , {  
*оператор* } , "  
 | TIMELINE , UNTIL ,  
*логічний\_вираз* , "{" , { *оператор* } , "  
 ;

Оператор послідовної обробки SEQUENCE – це специфічний оператор, який дозволяє об'єднувати дії з обробки даних, що виконуються

одночасно, в одну комплексну дію. Оператор послідовної обробки визначається синтаксичним правилом [SR3].

$$\begin{aligned} \text{оператор\_послідовної\_обробки} = & \text{SEQUENCE , "{" , { [SR3]} \\ & \text{оператор , [ ";" ]} , "}" ; \end{aligned}$$

Оператор розгалуження IF THEN (IF THEN ELSE) є стандартним оператором. Він має такий самий формат, як і в інших мовах програмування [336] та визначається синтаксичним правилом [SR4].

$$\begin{aligned} \text{оператор\_розгалуження} = & \text{IF , логічний\_вираз , THEN [SR4]} \\ & \text{, "{" , { оператор } , "}" | IF , логічний\_вираз ,} \\ & \text{THEN , "{" , { оператор } , "}" , ELSE , "{" , {} \\ & \text{оператор } , "}" ; \end{aligned}$$

Оператор вибору CASE OF (CASE OF ELSE) є стандартним оператором. Він має такий самий формат, як і в інших мовах програмування [336] та визначається синтаксичним правилом [SR5].

$$\begin{aligned} \text{оператор\_вибору} = & \text{CASE , ідентифікатор , OF , "{" , [SR5]} \\ & \text{{ ( ідентифікатор | значення ) , ":" , оператор } } \\ & \text{, "}" | CASE , ідентифікатор , OF , "{" , { (} \\ & \text{ідентифікатор | значення ) , ":" , оператор } ,} \\ & \text{"}" , ELSE , "{" , ( ідентифікатор | значення ) ,} \\ & \text{":" , оператор , "}" ; \end{aligned}$$

Оператор заміни SUBSTITUTE FOR WHEN є специфічним оператором, який підставляє один набір даних замість іншого при виконанні певної умови (наприклад, замінює дані високої роздільності

даними низької роздільності, якщо дані високої роздільності не можуть бути завантажені, вивантажені або оброблені через певну зовнішню причину, як-то низька швидкість передавання даних, відсутність зв'язку з зовнішнім джерелом даних тощо). Цей оператор визначається синтаксичним правилом [SR6].

*оператор\_заміни* = SUBSTITUTE , *ідентифікатор* , [SR6]  
FOR , *ідентифікатор* , WHEN *логічний\_вираз* ;

Оператор вивантаження DOWNLOAD FROM (DOWNLOAD FROM WITH) є специфічним оператором, який дозволяє вивантажувати дані з певного джерела даних (віддаленого чи локального, файлу чи потоку даних) та присвоювати ці дані певній змінній (елемента, кортежу або агрегата). Перетворення даних з файлу у формат змінної виконується з використанням певного обробника з відповідної бібліотеки. Оператор вивантаження визначається синтаксичним правилом [SR7].

*оператор\_вивантаження* = DOWNLOAD , [SR7]  
*ідентифікатор* , FROM , *ідентифікатор* [  
WITH , *ідентифікатор* ] ;

Оператор завантаження UPLOAD TO (UPLOAD TO WITH) – це специфічний оператор, який дозволяє вивантажувати дані, що присвоєні певній змінній (елемента, кортежу або агрегата) до певного ресурсу (віддаленого чи локального) для подальшого збереження як файл. Перетворення даних з формату змінної у файл виконується з використанням певного обробника з відповідної бібліотеки. Оператор завантаження визначається синтаксичним правилом [SR8].



*оператор\_завантаження* = UPLOAD , [SR8]  
*ідентифікатор* , TO , *ідентифікатор* [ WITH ,  
*ідентифікатор* ] ;

Оператор присвоєння IS або = є стандартним оператором. Він має наступний формат, що визначений синтаксичним правилом [SR9].

*оператор\_присвоювання* = *ідентифікатор*, ( IS | "=" ) [SR9]  
, *значення* ;

Оператор відтворення RENDER WITH є специфічним оператором, який дозволяє відтворювати дані за допомогою певного засобу відтворення. Він визначається синтаксичним правилом [SR10].

*оператор\_відтворення* = RENDER , *ідентифікатор* , [SR10]  
WITH , *ідентифікатор* ;

В мові ASAMPL використовуються математичні операції, визначені синтаксичним правилом [SR11].

*математична\_операція* = "+" | "-" | "\*" | "/" | "^" ; [SR11]

В мові ASAMPL використовуються логічні операції кон'юнкція, диз'юнкція та заперечення, визначені синтаксичними правилами [SR12]-[SR15].

*логічна\_операція* = *кон'юнкція* | *диз'юнкція* | [SR12]  
*заперечення* ;

*кон'юнкція* = "AND" | "&" ; [SR13]

*диз'юнкція* = "OR" | "|" ; [SR14]

*заперечення* = "NOT" | "~" ; [SR15]

Оскільки мова ASAMPL ґрунтується на апараті АСА, то в ній використовуються операції над агрегатами, кортежами та їх елементами, визначені в АСА, а саме, логічні операції та операції впорядкування.

Логічними операціями АСА є об'єднання, перетинання, різниця, симетрична\_різниця та виключний\_перетин. Ці операції визначаються синтаксичними правилами [SR16]-[SR21].

*логічна\_операція\_АСА* = *об'єднання* | *перетин* | [SR16]  
*різниця* | *симетрична\_різниця* |  
*виключний\_перетин* ;

*об'єднання* = "UNI" | "&&" ; [SR17]

*перетин* = "SEC" | "##" ; [SR18]

*різниця* = "DIF" | "\\\" ; [SR19]

*симетрична\_різниця* = "SDIF" | "^^" ; [SR20]

*виключний\_перетин* = "XSEC" | "%%" ; [SR21]

Операціями впорядкування АСА є розміщення, сортування, проріджування, видалення та вставлення. Ці операції визначаються синтаксичними правилами [SR22]-[SR27].

*операція\_впорядкування\_ACA* = *розміщення* | [SR22]  
*сортування* | *проріджування* | *видалення* |  
*вставлення* ;

*розміщення* = "ORD" | "|=" ; [SR23]

*сортування* = "ASORT" | "DSORT" | "->" | "<-" ; [SR24]

*проріджування* = "SINGL" | "||" ; [SR25]

*видалення* = "EXTR" | "<>" ; [SR26]

*вставлення* = "INS" | "><" ; [SR27]

Логічні операції та операції впорядкування реалізуються програмним шляхом згідно з формальним визначенням цих операцій, наведеним у розділі 2. При реалізації операції сортування доцільно застосувати алгоритм швидкого сортування, зокрема, MSD Radix Sort [338].

Узагальнено оператори, операції та операнди у мові ASAMPL визначаються синтаксичними правилами [SR28]-[SR31].

*оператор* = *часовий\_оператор* | [SR28]  
*оператор\_послідовної\_обробки* |  
*оператор\_розгалуження* | *оператор\_вибору* |  
*оператор\_заміни* | *оператор\_завантаження* |  
*оператор\_присвоювання* |  
*оператор\_відтворення* | *операнд* ,

*бінарна\_операція* , *операнд* | *унарна\_операція* ,  
*операнд* ;

*операнд* = *ідентифікатор* | *значення* ; [SR29]

*бінарна\_операція* = *математична\_операція* | [SR30]  
*кон'юнкція* | *диз'юнкція* | *логічна\_операція\_ASA*  
| *операція\_впорядкування\_ASA* ;

*унарна\_операція* = *заперечення* ; [SR31]

В мові ASAMPL дані можуть належати до однієї з таких категорій:

1. *Element*, елемент – це одиничне значення.
2. *Tuple*, кортеж – це впорядкований набір даних.
3. *Aggregate*, агрегат – це кортеж кортежів.

Дані кожної з цих категорій можуть належати до таких типів даних:

1. *ATIME* та *RTIME* – абсолютний час та відносний час відповідно.
2. *ADATE* та *RDATE* – абсолютна дата та відносна дата відповідно.
3. *INTEGER* – ціле число.
4. *REAL* – дійсне число.
5. *DOUBLE* – дійсне число подвійної точності.
6. *FRAME* – кадр, дані цього типу являють собою двовимірну матрицю значень інтенсивності кольору.
7. *TEXT* – текстовий тип
8. *LINK* – посилання або повне ім'я файлу, де зберігаються дані.
9. *LOGIC* – логічний тип.
10. *EVENT* – подія.

Типи даних визначаються синтаксичними правилами [SR32]-[SR43].

*тип* = *тип\_абсолютний\_час* | *тип\_відносний\_час* [SR32]  
| *тип\_абсолютна\_дата* | *тип\_відносна\_дата*  
| *чисельний\_тип* | *тип\_кадр* | *текстовий\_тип*  
| *тип\_посилання* | *логічний\_тип* | *тип\_подія* ;

*тип\_множини* = *тип* ; [SR33]

*тип\_абсолютний\_час* = ATIME ; [SR34]

*тип\_відносний\_час* = RTIME ; [SR35]

*тип\_абсолютна\_дата* = ADATE ; [SR36]

*тип\_відносна\_дата* = RDATE ; [SR37]

*чисельний\_тип* = INTEGER | REAL | DOUBLE ; [SR38]

*тип\_кадр* = FRAME ; [SR39]

*текстовий\_тип* = TEXT ; [SR40]

*тип\_посилання* = LINK ; [SR41]

*логічний\_тип* = LOGIC ; [SR42]

*тип\_подія* = EVENT ; [SR43]

Значення, залежно від типу даних, мають у мові ASAMPLE синтаксис, що визначений правилами [SR44]-[SR57]. Синтаксичні правила

[SR45]-[SR48] визначені згідно з форматами часу та дати, що визначені стандартом ДСТУ ISO 8601:2010 [339] та рекомендаціями W3C [340].

*значення* = *значення\_абсолютного\_часу* [SR44]  
 | *значення\_відносного\_часу* |  
*значення\_абсолютної\_дати* |  
*значення\_відносної\_дати* | *число* | *кадр* |  
*текстова\_послідовність* | *посилання* |  
*логічне\_значення* | *подія* ;

*значення\_абсолютного\_часу* = [ *цифра* , { *цифра* } , [SR45]  
 ":" , ] *цифра* , [ *цифра* ] , ":" , *цифра* , [ *цифра* ] ,  
 [ ":" , *цифра* , [ *цифра* ] , "." , *цифра* ] , [ ("+" | "-"  
 ") , *цифра* , *цифра* , ":" , *цифра* , *цифра* ] ;

*значення\_відносного\_часу* = [ ("+" | "-") ] , [ *цифра* , { [SR46]  
*цифра* } , ":" ] *цифра* , { *цифра* } , ":" , *цифра* , {  
*цифра* } [ , ":" , *цифра* , { *цифра* } ] ;

*значення\_абсолютної\_дати* = [ *цифра* , *цифра* , [ [SR47]  
*цифра* , *цифра* ] , "/" , ] *цифра* , [ *цифра* ] , "/" ,  
*цифра* [ , *цифра* ] ;

*значення\_відносної\_дати* = [ ("+" | "-") ] , [ *цифра* , [SR48]  
*цифра* , { *цифра* , *цифра* } , "/" , ] *цифра* , {  
*цифра* } , "/" , *цифра* , { *цифра* } ;

*число* = *ціле\_число* | *дійсне\_число* | [SR49]  
*дійсне\_число\_подвійної\_точності* ;

*ціле\_число* = [ "-" ] , *цифра* , { *цифра* } ; [SR50]

*дійсне\_число* = [ "-" ] , *цифра* , { *цифра* } , "." , *цифра* , [SR51]  
{ *цифра* } ;

*дійсне\_число\_подвійної\_точності* = [ "-" ] , *цифра* , [SR52]  
"." , *цифра* , { *цифра* } , "e" , ( "-" | "+" ) , *цифра* ,  
{ *цифра* } ;

*кадр* = "[" , *число* , { "," , *число* } , { ";" , *число* , { "," , [SR53]  
*число* } } "]" ;

*текстова\_послідовність* = "" , { *літера* | *символ* | [SR54]  
*цифра* } , "" ;

*посилання* = "" , *літера* , *літера* , *літера* , [ *літера* | [SR55]  
*символ* | *цифра* ] "://" , ( *літера* | *символ* | *цифра*  
) { *літера* | *символ* | *цифра* } , "." , ( *літера* |  
*символ* | *цифра* ) { *літера* | *символ* | *цифра* } , {  
"/" , ( *літера* | *символ* | *цифра* ) { *літера* | *символ*  
| *цифра* } , "." , ( *літера* | *символ* | *цифра* ) {  
*літера* | *символ* | *цифра* } } , "" | "" , *літера* ,  
":\" , ( *літера* | *символ* | *цифра* ) { *літера* | *символ*  
| *цифра* } , { "\" , ( *літера* | *символ* | *цифра* ) {  
*літера* | *символ* | *цифра* } } , "" ;

*логічне\_значення* = TRUE | FALSE ; [SR56]

*подія* = ідентифікатор ; [SR57]

Інші базові поняття, що визначені у граматиці мови ASAMPL, представлені синтаксичними правилами [SR58]-[SR65].

*літера* = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" [SR58]  
| "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" |  
"S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z" | "a" |  
"b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l"  
| "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" |  
"w" | "x" | "y" | "z" ;

*цифра* = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ; [SR59]

*символ* = "." | "," | ";" | ":" | "!" | "?" | "(" | ")" | "[" | "]" | [SR60]  
"{" | "}" | "<" | ">" | "\"" | "\"" | "+" | "-" | "\*" | "/" |  
"=" | "\" | "@" | "#" | "\$" | "|" | "%" | "^" | "&" ;

*ідентифікатор* = літера , { літера | цифра } ; [SR61]

*кортеж\_значень* = ідентифікатор | "[" , { значення [ [SR62]  
", " ] } , "]" ;

*кортеж\_значень\_часу* = ідентифікатор | "[" , { [SR63]  
значення\_часу [ ", " ] } , "]" ;



*логічний\_вираз* = *ідентифікатор* | *ідентифікатор* , [SR64]  
*логічна\_операція* , *ідентифікатор* |  
*логічне\_значення* ;

*коментар* = *"//"* , *текстова\_послідовність* ; [SR65]

Визначимо семантику основних елементів мови ASAMPL.

#### 4.4.2. Семантика мови ASAMPL

Для визначення семантики операторів мови ASAMPL використовуватимемо аксіоматичний та операційний підходи.

Аксіоматичні специфікації операторів визначатимемо як правила виведення [341, 342] виду:

$$\frac{H_1, H_2, \dots, H_n}{H} ,$$

де  $H$  є вірним, якщо  $\forall H_i$  є вірним (верифікованим),  $i = 1..n$ .

Для визначення операційної семантики операторів, для яких немає аналогів у інших мовах програмування, будемо використовувати діаграми станів UML [343, 344].

Аксіоматична семантика операторів **TIMELINE**, **TIMELINE AS** та **TIMELINE UNTIL** синтаксис яких визначений правилом [SR2], визначається формулами [SF1]-[SF3], де  $P$  та  $Q$  визначають попереднє та наступне твердження,  $C_1, C_2, \dots, C_K$  визначають дії, що потрібно одночасно виконати у часовому операторі,  $a_1$  та  $a_2$  визначають час початку та кінця виконання часового оператора,  $\Delta a$  – крок зміни значень часу,  $[a_1..a_n]$  – кортеж часових значень,  $x$  – поточне значення часу,  $B$  – умова, яка визначає завершення процесу виконання дій у операторі.

$$\frac{\{P \& (a_1 \leq x \leq a_2)\} C_1, C_2, \dots, C_K \{Q\}}{\{P\} \text{TIMELINE } (a_1 : \Delta a : a_2) C_1, C_2, \dots, C_K \{Q\}} \quad [\text{SF1}]$$

$$\frac{\{P \& (x \in [a_1..a_n])\} C_1, C_2, \dots, C_K \{Q\}}{\{P\} \text{TIMELINE AS } [a_1..a_n] C_1, C_2, \dots, C_K \{Q\}} \quad [\text{SF2}]$$

$$\frac{\{P \& B\} C_1, C_2, \dots, C_K \{Q\}}{\{P\} \text{TIMELINE UNTIL } B C_1, C_2, \dots, C_K \{Q\}} \quad [\text{SF3}]$$

Операційна семантика оператора TIMELINE визначається діаграмою станів, яка наведена на рис. 4.3.

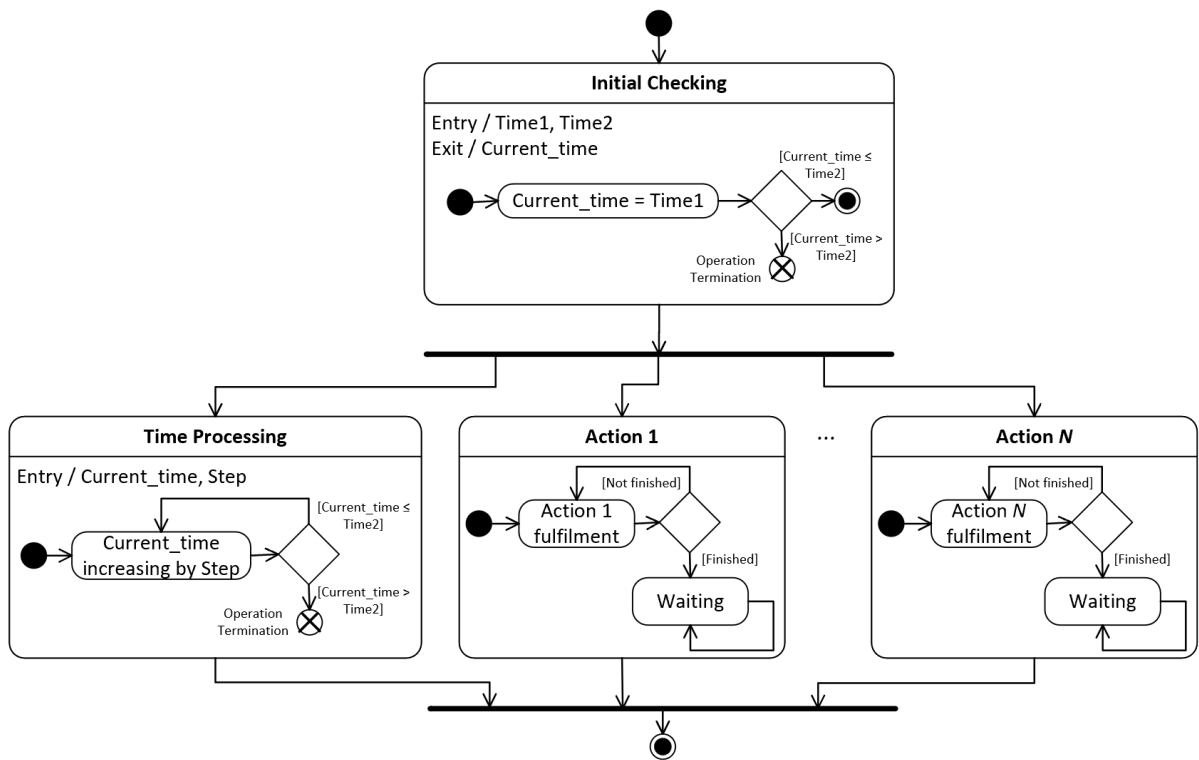


Рис. 4.3. Діаграма станів оператора TIMELINE

Операційна семантика оператора TIMELINE AS визначається діаграмою станів, яка наведена на рис. 4.4.

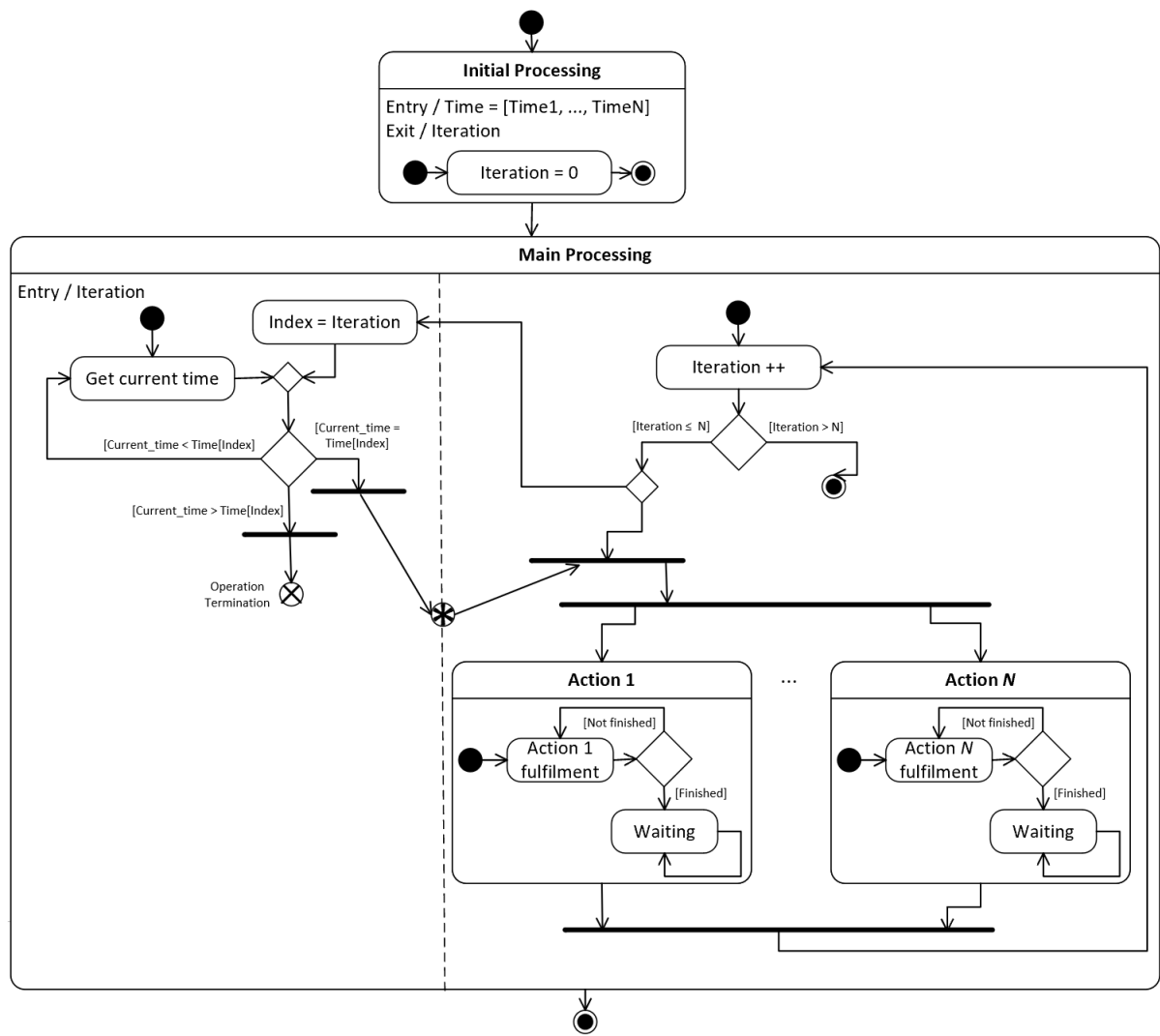


Рис. 4.4. Діаграма станів оператора TIMELINE AS

Операційна семантика оператора TIMELINE UNTIL визначається діаграмою станів, яка наведена на рис. 4.5.

Аксіоматична семантика оператора SEQUENCE, синтаксис якого визначений правилом [SR3], визначається формулою [SF4], де  $P$  та  $R$  визначають попереднє та наступне твердження відповідно,  $Q_1, \dots, Q_{K-1}$  визначають проміжні твердження,  $C_1, C_2, \dots, C_K$  визначають дії, що потрібно послідовно виконати у операторі SEQUENCE.

$$\frac{\{P\} C_1 \{Q_1\}, \{Q_1\} C_2 \{Q_2\}, \dots, \{Q_{K-1}\} C_K \{R\}}{\{P\} \text{SEQUENCE } C_1, C_2, \dots, C_K \{R\}} \quad [\text{SF4}]$$

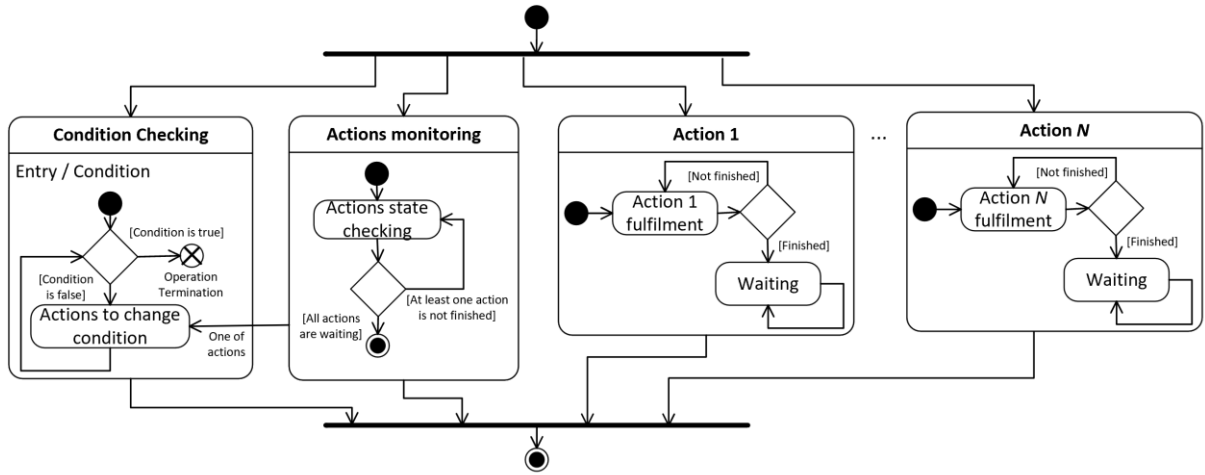


Рис. 4.5. Діаграма станів оператора TIMELINE UNTIL

Операційна семантика оператора SEQUENCE визначається діаграмою станів, яка наведена на рис. 4.6.

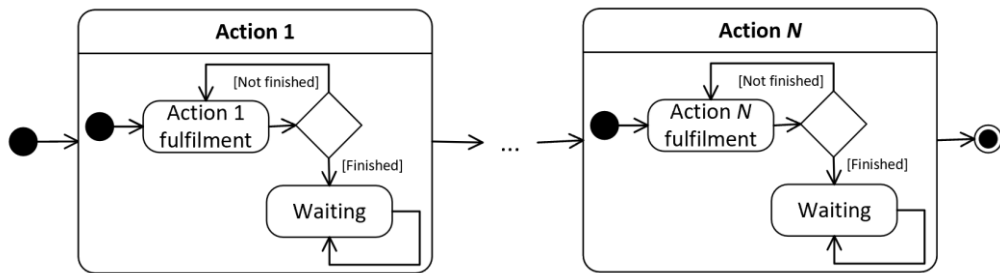


Рис. 4.6. Діаграма станів оператора SEQUENCE

Аксіоматична семантика оператора IF, синтаксис якого визначений правилом [SR4], визначається формулами [SF5] та [SF6], де  $P$  та  $Q$  визначають попереднє та наступне твердження відповідно,  $C, C_1, C_2$  визначають альтернативні дії,  $B$  – умова вибору альтернативних дій для виконання у операторі IF.

$$\frac{\{P \& B\} C \{Q\}, (P \& \bar{B}) \supset Q}{\{P\} \text{ IF } B \text{ THEN } C \{Q\}} \quad [\text{SF5}]$$

$$\frac{\{P \& B\} C_1 \{Q\}, \{P \& \bar{B}\} C_2 \{Q\}}{\{P\} \text{ IF } B \text{ THEN } C_1 \text{ ELSE } C_2 \{Q\}} \quad [\text{SF6}]$$

Аксіоматична семантика оператора CASE, синтаксис якого визначений правилом [SR5], визначається формулами [SF7] та [SF8], де  $P$  та  $Q$  визначають попереднє та наступне твердження відповідно,  $k_i$  визначають номери альтернативних варіантів дій та  $C_i$  визначають відповідні альтернативні дії ( $i = 1..n$ ),  $x$  визначає поточний номер варіанту.

$$\frac{\{P \& (x = k_i \mid i = 1..n)\} C_i \{Q\}}{\{P \& (x = k_i)\} \text{ CASE } x \text{ OF } k_i : C_i \{Q\}} \quad [\text{SF7}]$$

$$\frac{\{P \& (x = k_i \mid i = 1..n)\} C_i \{Q\}, \{P \& (x \notin [k_1 \dots k_n])\} C \{Q\}}{\{P \& (x = k_i)\} \text{ CASE } x \text{ OF } k_i : C_i \text{ ELSE } C \{Q\}} \quad [\text{SF8}]$$

Аксіоматична семантика оператора SUBSTITUTE, синтаксис якого визначений правилом [SR6], визначається формулою [SF9], де  $P$  та  $Q$  визначають попереднє та наступне твердження відповідно,  $C_1, C_2$  визначають дії над даними,  $s_1, s_2$  визначають джерела даних,  $d$  визначає змінну, з якою асоціюються дані,  $B$  – умова вибору джерела даних.

$$\frac{\{P \& B\} C_1 \{Q\}, \{P \& \bar{B}\} C_2 \{Q\}, C_i \equiv (d = d_i)}{\{P\} d \text{ SUBSTITUTE } s_2 \text{ FOR } s_1 \text{ WHEN } B \{Q\}} \quad [\text{SF9}]$$

Операційна семантика оператора SUBSTITUTE визначається діаграмою станів, яка наведена на рис. 4.7.

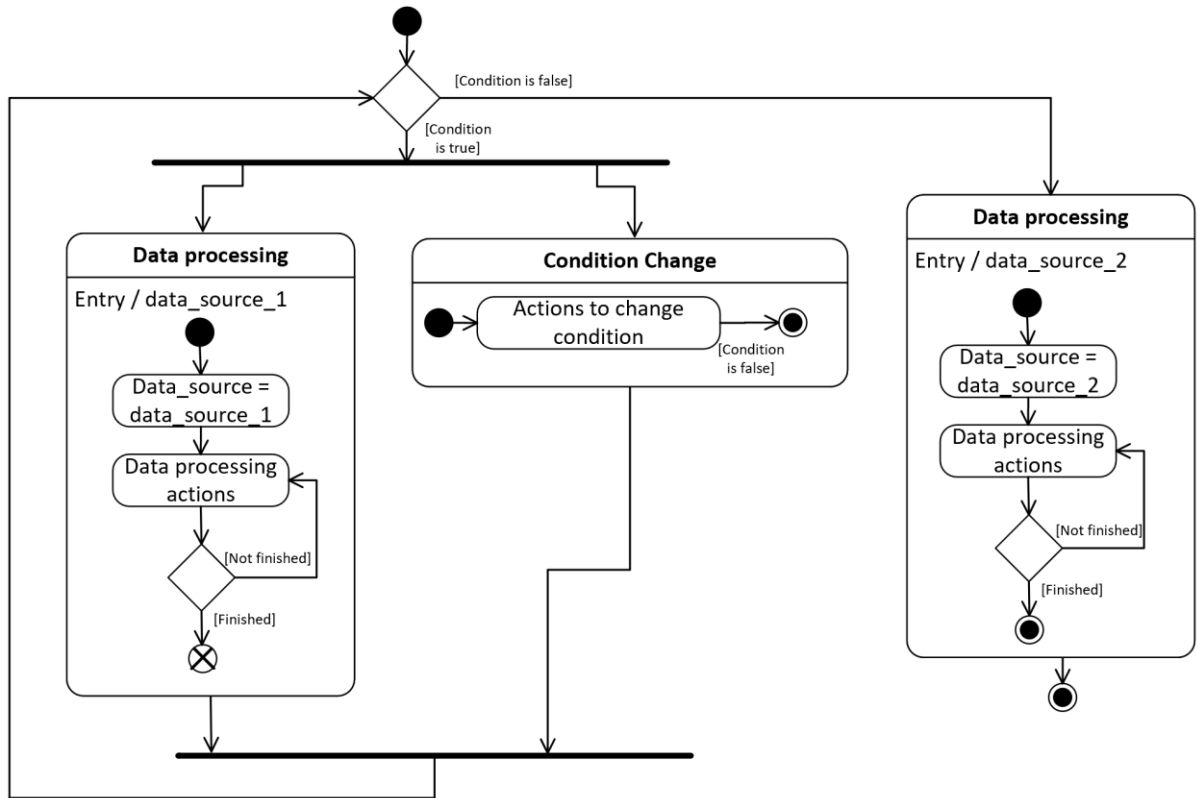


Рис. 4.7. Діаграма станів оператора SUBSTITUTE

Аксіоматична семантика оператора DOWNLOAD, синтаксис якого визначений правилом [SR7], визначається формулами [SF10] та [SF11], де  $P$  та  $Q$  – попереднє та наступне твердження відповідно,  $C$  визначає дію над даними,  $s$  визначає джерело даних,  $d$  визначає змінну, з якою асоціюються дані,  $z$  визначає процедуру конвертації даних (обробник).

$$\frac{\{P\} C \{Q\}}{\{P\} \text{DOWNLOAD } d \text{ FROM } s \{Q\}} \quad [\text{SF10}]$$

$$\frac{\{P\} C \{Q\}}{\{P\} \text{DOWNLOAD } d \text{ FROM } s \text{ WITH } z \{Q\}} \quad [\text{SF11}]$$

Операційна семантика оператора DOWNLOAD визначається діаграмою станів, яка наведена на рис. 4.8.

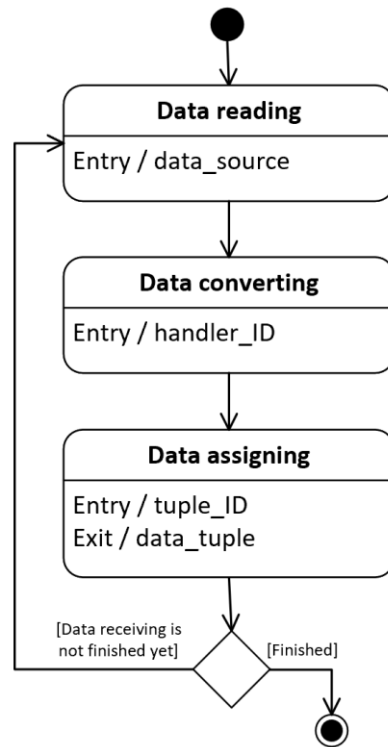


Рис. 4.8. Діаграма станів оператора DOWNLOAD

Аксиоматична семантика оператора UPLOAD, синтаксис якого визначений правилом [SR8], визначається формулами [SF12] та [SF13], де  $P$  та  $Q$  – попереднє та наступне твердження відповідно,  $C$  визначає дію над даними,  $r$  визначає приймач даних,  $d$  визначає змінну, з якою асоціюються дані,  $z$  визначає процедуру конвертації даних (обробник).

$$\frac{\{P\} C \{Q\}}{\{P\} \text{UPLOAD } d \text{ TO } r \{Q\}} \quad [\text{SF12}]$$

$$\frac{\{P\} C \{Q\}}{\{P\} \text{UPLOAD } d \text{ TO } r \text{ WITH } z \{Q\}} \quad [\text{SF13}]$$

Операційна семантика оператора UPLOAD визначається діаграмою станів, яка наведена на рис. 4.9.

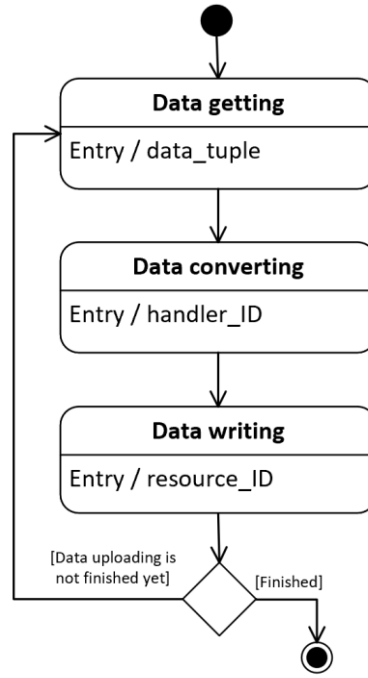


Рис. 4.9. Діаграма станів оператора UPLOAD

Аксіоматична семантика оператора IS, синтаксис якого визначений правилом [SR9], визначається формулою [SF14], де  $P$  та  $E$  – твердження,  $x$  – змінна,  $P[E/x]$  означає результат підстановки виразу  $E$  замість всіх входжень змінної  $x$  у вираз  $P$  [341].

$$\{P[E/x]\} x \text{ IS } E \{P\} \quad [\text{SF14}]$$

Аксіоматична семантика оператора RENDER, синтаксис якого визначений правилом [SR10], визначається формулою [SF15], де  $P$  та  $Q$  – попереднє та наступне твердження відповідно,  $C$  визначає дію над даними,  $d$  визначає змінну, з якою асоціюються дані,  $v$  визначає процедуру підготовки даних до відтворення (рендер).

$$\frac{\{P\} C \{Q\}}{\{P\} \text{ RENDER } d \text{ WITH } v \{Q\}} \quad [\text{SF15}]$$



Операційна семантика оператора RENDER визначається діаграмою станів, яка наведена на рис. 4.10.

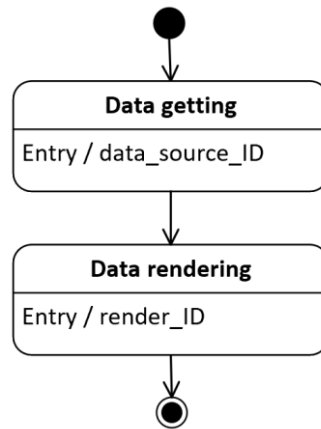


Рис. 4.10. Діаграма станів оператора RENDER

Розглянемо семантику типів даних, що визначені синтаксичними правилами [SR45]-[SR48].

Нехай  $a, b \in T_A$ , де  $T_A$  – множина значень типу ATIME, на якій задані бінарні операції  $+$  (додавання),  $-$  (віднімання) та відношення  $>$  (перевищує),  $<$  (не перевищує),  $=$  (дорівнює),  $>$  (слідуює),  $<$  (передуює),  $\equiv$  (співпадає). Тоді справедливими є наступні твердження:

1.  $f(a, b) = a + b, f: T_A \times T_A \rightarrow T_A$
2.  $a + b = b + a$
3.  $f(a, b) = a - b, f: T_A \times T_A \rightarrow T_A$
4.  $|a - b| = |b - a|$
5.  $(0 < a < b) \equiv (a < b) \equiv [a, b]$
6.  $(b > a > 0) \equiv (b > a) \equiv [a, b]$
7.  $(a = b) \equiv (a \equiv b) \equiv [a] \equiv [b]$ .

Нехай  $a, b \in T_R$ , де  $T_R$  – множина значень типу RTIME, на якій задані бінарні операції  $+$  (додавання),  $-$  (віднімання) та відношення

$>$  (перевищує),  $<$  (не перевищує),  $=$  (дорівнює),  $\succ$  (слідуює),  $\prec$  (передуює),  $\equiv$  (співпадає). Тоді справедливими є наступні твердження:

1.  $f(a, b) = a + b, f: T_R \times T_R \rightarrow T_R$
2.  $a + b = b + a$
3.  $f(a, b) = a - b, f: T_R \times T_R \rightarrow T_R$
4.  $|a - b| = |b - a|$
5.  $(a < b) \equiv (a \prec b) \equiv [a, b]$
6.  $(b > a) \equiv (b \succ a) \equiv [a, b]$
7.  $(a = b) \equiv (a \equiv b) \equiv [a] \equiv [b]$ .

Нехай  $a, b \in D_A$ , де  $D_A$  – множина значень типу ADATE, на якій задані бінарні операції  $+$  (додавання),  $-$  (віднімання) та відношення  $>$  (перевищує),  $<$  (не перевищує),  $=$  (дорівнює),  $\succ$  (слідуює),  $\prec$  (передуює),  $\equiv$  (співпадає). Тоді справедливими є наступні твердження:

1.  $f(a, b) = a + b, f: D_A \times D_A \rightarrow D_A$
2.  $a + b = b + a$
3.  $f(a, b) = a - b, f: D_A \times D_A \rightarrow D_A$
4.  $|a - b| = |b - a|$
5.  $(0 < a < b) \equiv (a \prec b) \equiv [a, b]$
6.  $(b > a > 0) \equiv (b \succ a) \equiv [a, b]$
7.  $(a = b) \equiv (a \equiv b) \equiv [a] \equiv [b]$ .

Нехай  $a, b \in D_R$ , де  $D_R$  – множина значень типу RDATE, на якій задані бінарні операції  $+$  (додавання),  $-$  (віднімання) та відношення  $>$  (перевищує),  $<$  (не перевищує),  $=$  (дорівнює),  $\succ$  (слідуює),  $\prec$  (передуює),  $\equiv$  (співпадає). Тоді справедливими є наступні твердження:

1.  $f(a, b) = a + b, f: D_R \times D_R \rightarrow D_R$
2.  $a + b = b + a$

3.  $f(a, b) = a - b, f: D_R \times D_R \rightarrow D_R$
4.  $|a - b| = |b - a|$
5.  $(a < b) \equiv (a < b) \equiv [a, b]$
6.  $(b > a) \equiv (b > a) \equiv [a, b]$
7.  $(a = b) \equiv (a \equiv b) \equiv [a] \equiv [b]$ .

Розглянемо застосування конструкцій мови ASAMPL для реалізації моделі обчислень парадигми програмування мультимедіа.

#### 4.4.3. Прагматика мови ASAMPL

Програмування мовою ASAMPL є відносно простим процесом завдяки наявності у мові операторів, які дозволяють розробнику виконувати базові операції над мультимодальними даними: синхронізовувати дані різних модальностей, динамічно їх замінювати та завантажувати. Особливу увагу у мові приділено такому типу даних як часові дані. Час є особливою сутністю для мультимодальних даних, оскільки всі мультимедійні та мультимедійні дані є темпоральними. Обчислювальна модель парадигми програмування для обробки темпоральних мультимодальних даних може бути реалізована мовою програмування ASAMPL шляхом визначення агрегатів темпоральних мультимодальних даних з синхронізацією потоків даних із застосуванням часового оператора.

Мультимедіа на першому етапі синхронізації та агрегації визначається у розділі AGGREGATES агрегатом, як показано у лістингу 4.8.

Лістинг 4.8 – Визначення часткового мультимедіа мовою ASAMPL

```
MultiImage_i = [time, data_modality_i];
```

Мультиобраз на другому етапі базової моделі обчислень визначається у розділі AGGREGATES агрегатом виду, як показано у лістингу 4.9.

Лістинг 4.9 – Визначення мультиобразу об'єкта мовою ASAMPL

```
MultiImage = [time, data_modality_1, ...,  
data_modality_N];
```

Процедура синхронізації визначається відповідно до визначених правил та шаблонів синхронізації мультимодальних даних та реалізується із застосуванням оператора TIMELINE, як показано у лістингу 4.10.

Лістинг 4.10 – Реалізація процедури синхронізації мовою ASAMPL

```
TIMELINE time1 : step : time2 {  
    ... // synchronization rules realization }
```

Таким чином, парадигма програмування мультиобразів може бути реалізована за допомогою мови програмування ASAMPL.

Можливі застосування даної мови програмування належать до сфери освіти, медицини, телекомунікаційних систем, промислових систем, систем моніторингу, електронної комерції, програмно-апаратних систем IoT тощо.

Розглянемо приклад програмного коду, який призначено для обробки та формування мультимедійного контенту для уроку географії (лістинг 4.11).

Лістинг 4.11 – Приклад програмного коду мовою ASAMPL

```
PROGRAM GeographyLesson {  
    LIBRARIES {  
        VisualLib Is 'D:\MImage\Library\vsl.lib';  
        AudioLab Is 'D:\MImage\Library\aud.lib';  
        OlfactLib Is 'D:\MImage\Library\olf.lib';  
    }
```

```

HANDLERS {
    MPEG2tuple Is 'D:\MImage\Handl\mpg2tup.exe';
    Tuple2MPEG Is 'D:\MImage\Handl\tup2mpg.exe';}
RENDERERS {
    VisualRen Is 'C:\Renderer\vslren.exe';
    AudioRen Is 'C:\Renderer\audren.exe';
    OlfactRen Is 'C:\Renderer\olfren.exe';}
SOURCES {
    VideoFile Is
`http://edu.net/Forest&WildLife.mp4';
    VisualDataStream Is
`http://webcam.edu.net/005201';
    OlfactFile Is 'D:\Lesson05\forest.dat';
    SceneFile Is 'D:\Lesson05\forest.agg';}
SETS {
    Frame Is Integer(1920,1080);
    Audio Is Real;
    Scent Is Text;}
ELEMENTS {
    duration is Time;
    time1 = 00:00:01;
    time2 = 00:15:00;
    step = 00:00:01;}
TUPLES {
    VisualDat Is Frame;
    VisualDat2 Is Frame;
    AudioDat Is Audio;
    OlfactDat Is Scent;}
AGGREGATES {
    ForestScene = [VisualDat,AudioDat,OlfactDat];}
ACTIONS {
    DOWNLOAD AudioDat FROM VideoFile.audio
        WITH MPEG2tuple;
    DOWNLOAD VisualDat2 FROM VideoFile.visual
        WITH MPEG2tuple;
    DOWNLOAD OlfactDat FROM OlfactFile
        WITH default.OlfactLib;
    TIMELINE time1 : step : time2 {

```

```

DOWNLOAD VisualDat FROM VisualDataStream
    WITH default.VisualLib;}
SUBSTITUTE VisualDat2 FOR VisualDat
    WHEN VisualDataStream IS Equivalent Null;
UPLOAD ForestScene TO SceneFile WITH
default.all;
RENDER ForestScene
    WITH [VisualRen, AudioRen, OlfactRen];} }

```

Цей програмний код дозволяє створювати віртуальну модель (мультиобраз) реальної сцени. Мультиобраз включає дані трьох модальностей: відео, аудіо та запах. Передбачено два можливих джерела відео-даних: з файлу та з веб-камери, яка знаходиться у певному місті лісного масиву та зв'язок з якою встановлюється через мережу. Передбачається, що веб-камера може бути вимкнена з певних причин. Ця технічна проблема не має впливати на хід проведення шкільного уроку з географії, тому у коді програми використовується оператор *Substitute For When*. Він дозволяє підставляти дані з файлу, записаного для цього уроку заздалегідь, замість реального потоку відео з веб-камери, що транслює зображення лісної сцени у реальному часі. Часова шкала мультиобразу розрахована на 15 хвилин. Протягом цього часу потік візуальних даних спочатку завантажується та представляється у програмі як кортеж. Цей кортеж може додатково оброблюватись, але у цьому прикладі він використовується як складова мультиобразу, що у програмі представлений як агрегат. Цей агрегат зберігається у вигляді файлу формату *.agg* на локальному диску та відтворюється комплексним рендером, який складається з трьох рендерів, кожен з яких відтворює дані певної модальності.

Практична реалізація відтворення мультимедійних даних мультиобразу може бути досягнута використанням відповідних пристроїв за технологією мультимедіа [193–206, 208–210, 212–215, 217].

#### 4.4.4. Порівняльний аналіз мови ASAMPL

Порівняємо реалізацію базових операцій над мультимодальними даними у мові ASAMPL та в універсальних мовах програмування.

Паралельна обробка темпоральних мультимодальних даних виконується у мові ASAMPL за допомогою оператора `TIMELINE`, як показано у лістингу 4.12.

Лістинг 4.12 – Паралельна обробка темпоральних мультимодальних даних мовою ASAMPL

```
TIMELINE time1 : step : time2
{  action1;
   ...
   actionN; }
```

В універсальних мовах програмування аналогічна обробка виконується за допомогою декількох конструкцій. Наприклад, фрагмент коду мовою C# наведено у лістингу 4.13.

Лістинг 4.13 – Паралельна обробка темпоральних мультимодальних даних мовою C#

```
class Timeline
{  List<Action> actions = new List<Action>();
   private double timePeriod;
   public Timeline(DateTime startTime, DateTime
   endTime, List<Action> actions)
   {  timePeriod = (endTime -
       startTime).TotalMilliseconds;
       this.actions = actions; }
   public async Task TimelineAsync(int timeStep)
   {  List<Task> listOfTasks = new List<Task>();
       foreach (var action in actions)
```

```

        listOfTasks.Add(DoAsync(action));
        await Task.WhenAll(listOfTasks); }
private Task DoAsync(Action action)
{
    Task.Delay(Convert.ToInt32(timePeriod));
    action();
    return Task.CompletedTask; } }

```

Отримання даних із довільного зовнішнього джерела, такого як файл у локальному або хмарному сховищі, потокові дані з веб-камери або давача тощо, виконується у мові ASAMPL за допомогою оператора DOWNLOAD, як показано у лістингу 4.14.

Лістинг 4.14 – Отримання темпоральних мультимодальних даних із зовнішнього джерела мовою ASAMPL

```

DOWNLOAD name FROM link WITH handler;

```

В універсальних мовах програмування отримання даних із довільного зовнішнього джерела потребує більшої кількості дій. Наприклад, фрагмент коду мовою C# наведено у лістингу 4.15.

Лістинг 4.15 – Отримання темпоральних мультимодальних даних із зовнішнього джерела мовою C#

```

public void Download<T>(Stream inputData, string
link, string name)
{
    DirectoryInfo source = new
    DirectoryInfo(link);
    if (!source.Exists) source.Create();
    string path = Path.Combine(link, name);
    using FileStream outputData = new
    FileStream(path, FileMode.Create);
    inputData.CopyTo(outputData);
    Handler<T>(outputData); }

```



Заміна одного набору темпоральних мультимодальних даних іншим, наприклад, під час трансляції відео із забезпеченням динамічної зміни активної веб-камери у системах моніторингу, виконується у мові ASAMPL за допомогою оператора SUBSTITUTE, як показано у лістингу 4.16.

Лістинг 4.16 – Динамічна заміна джерела темпоральних мультимодальних даних мовою ASAMPL

```
SUBSTITUTE name2 FOR name1 WHEN condition ;
```

В універсальних мовах програмування динамічна заміна джерела темпоральних мультимодальних даних потребує застосування набору операцій. Наприклад, фрагмент коду мовою C# наведено у лістингу 4.17.

Лістинг 4.17 – Динамічна заміна джерела темпоральних мультимодальних даних мовою C#

```
class Substitute
{
    DataProcessing dataProcessing = new
        DataProcessing();
    public void SwitchingDataSource(string
        sourceOne, string sourceTwo, bool condition)
    {
        if (!condition)
            dataProcessing.ProcessData(sourceTwo);
        else
        {
            if (SubstituteAsync(sourceOne).Result)
                dataProcessing.ProcessData(sourceTwo);
        }
    }
    private async Task<bool>
        SubstituteAsync(string source)
    {
        CancellationTokenSource cts = new
            CancellationTokenSource();
        CancellationToken token = cts.Token;
        var processData = Task.Run(() =>
            dataProcessing.ProcessData(source, token));
    }
}
```

```

var conditionChange = Task.Run(() =>
    dataProcessing.ChangeCondition(token));
List<Task> tasks = new List<Task>()
{    processData, conditionChange };
var finishedTask = await
Task.WhenAny(tasks);
tasks.Remove(finishedTask);
if (finishedTask == processData)
{    cts.Cancel();
    return false; }
else
{    cts.Cancel();
    return true; } } }

```

Завантаження темпоральних мультимодальних даних до довільного сховища виконується у мові ASAMPL за допомогою оператора UPLOAD, як показано у лістингу 4.18.

Лістинг 4.18 – Завантаження темпоральних мультимодальних даних до сховища мовою ASAMPL

```

UPLOAD name TO link WITH handler;

```

В універсальних мовах програмування завантаження даних до довільного сховища потребує більшої кількості дій. Наприклад, фрагмент коду мовою C# наведено у лістингу 4.19.

Лістинг 4.19 – Завантаження темпоральних мультимодальних даних до сховища мовою C#

```

public void Upload<T>(string source, string name)
{    var dataTuple = DataReceiver<T>();
    var data = Handler<T>(dataTuple);
    string path = Path.Combine(source, name);

```

```
using StreamWriter writer = new
StreamWriter(path);
foreach (var dataPiece in data)
    writer.WriteLine(dataPiece); }
```

Проаналізуємо ефективність обробки темпоральних мультимодальних даних згідно з парадигмою програмування мультиобразів, зокрема, за допомогою універсальних мов програмування та за допомогою мови ASAMPL.

Для аналізу складності програмного коду обчислимо такі показники:

- цикломатична складність,
- кількість рядків програмного коду,
- індекс зручності підтримки.

*Цикломатична складність* (цикломатичне число Маккейба) відноситься до метрик складності потоку керування програм. Цей показник дозволяє оцінити структурну складність програмного коду шляхом визначення кількості можливих варіантів проходження по програмному коду при виконанні програми [345].

Для визначення цикломатичної складності для певного фрагменту коду потрібно побудувати для нього граф керуючої логіки. Граф керуючої логіки є орієнтованим графом, вузли якого визначають елементарні керуючі конструкції, з яких складається цей програмний код. Два вузли у графі керуючої логіки з'єднуються ребром, якщо відповідні цим вузлам керуючі конструкції виконуються послідовно. Наприклад, граф керуючої логіки для методу *SwitchingDataSource* (лістинг 4.17) наведено на рис. рис. 4.11.

Цикломатична складність може бути визначена за формулою (4.1).

$$\Psi_c = v_{edge} - v_{node} + 2 \cdot v_{comp} \quad (4.1)$$

де  $v_{edge}$  – кількість ребер;  $v_{node}$  – кількість вузлів;  $v_{comp}$  – кількість компонентів зв'язності.

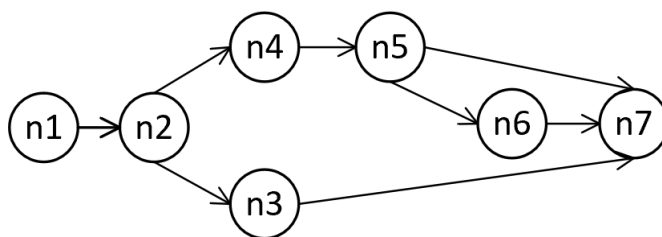


Рис. 4.11. Приклад графа керуючої логіки

*Індекс зручності підтримки* дозволяє оцінити, наскільки легким для підтримки та зміни є програмний код [346]. Існує декілька підходів до обчислення цього показника. Один з них визначається формулою (4.2).

$$\Psi_m = \max\left(0, 100 \cdot \frac{171 - 5.2 \cdot \ln V - 0.23 \cdot \Psi_c - 16.2 \cdot \ln L}{171}\right) \quad (4.2)$$

де  $V$  – обсяг програмного коду,

$L$  – кількість рядків програмного коду.

Обсяг програмного коду є одним з показників метрик Холстеда, відносяться до метрик розміру програм. Вони ґрунтуються на вимірюваних властивостях програмного коду та дозволяються встановити зв'язок між ними [347]. Базовими показниками метрик Холстеда є наступні:

- загальна кількість операторів ( $\eta_o$ ),
- загальна кількість операндів ( $\eta_d$ ),
- кількість унікальних операторів ( $\eta_{uo}$ ),
- кількість унікальних операндів ( $\eta_{ud}$ ).

Виходячи з цих базових показників можна визначити інші показники, зокрема, обсяг програмного коду, який обчислюється згідно з формулою (4.3).

$$V = (\eta_o + \eta_d) \cdot \log_2(\eta_{uo} + \eta_{ud}). \quad (4.3)$$

Тоді усереднені показники програмного коду, що реалізує управляючі конструкції, для мови програмування ASAMPL та універсальних мов програмування C#, C++ та Python наведені у табл. 4.2.

Таблиця 4.2 – Усереднені показники програмного коду

Мова програмування	C#	C++	Python	ASAMPL
Цикломатична складність	3.25	3.25	4	1
Обсяг програмного коду	152.94	171.60	131.52	14.53
Індекс зручності підтримки	54.16	55.26	62.35	89.15

Таким чином, програмний код спеціалізованою мовою ASAMPL має кращі показники у порівнянні з кодом, розробленим з використанням універсальних мов програмування.

#### 4.5. Висновки до розділу 4

У цьому розділі вирішувалась наукова задача підвищення ефективності програмної обробки темпоральних мультимодальних даних цифрових двійників. В результаті отримано такі результати.

1. Розроблено парадигму програмування мультиобrazів для обробки темпоральних мультимодальних даних. Особливостями цієї парадигми є:

- використання мультиобразу досліджуваного об'єкта як основного компонента обчислювального процесу, що дозволяє спростити технологію розроблення програмного забезпечення систем обробки темпоральних мультимодальних даних;

- застосування апарату алгебраїчної системи агрегатів для обробки структур темпоральних мультимодальних даних;
- використання синхронізації та агрегації даних різних модальностей як основної складової моделі обчислень;
- орієнтація на використання різномірних джерел даних.

2. Розроблено спеціалізовану мову програмування ASAMPL, яка реалізує запропоновану парадигму програмування. Мова програмування ASAMPL призначена для розроблення прикладного програмного забезпечення систем обробки темпоральних мультимодальних даних, зокрема мультимедійних та мультисемедійних застосунків. Характерними рисами мови програмування ASAMPL є:

- обробка мультимодальних даних ґрунтується на використанні спеціальних структур даних – мультобразів, які реалізуються за допомогою агрегатів та кортежів;
- для обробки темпоральних мультимодальних даних використовуються спеціальні оператори: часовий оператор, оператор заміни, оператори вивантаження та завантаження, оператор відтворення.

3. Застосування спеціальних операторів мови ASAMPL для обробки темпоральних мультимодальних даних дозволяє зменшити обсяг програмного коду у 4-6 разів, зменшити цикломатичну складність програми у 2-3 рази та підвищити індекс зручності підтримки програмного коду на 25-30% порівняно з мовами програмування C#, C++, C та Python.

4. Запропонована парадигма програмування дозволяє підвищити ефективність процесу розроблення прикладного програмного забезпечення технології цифрових двійників за рахунок зменшення складності програмного коду та поліпшення зручності його обслуговування.

## **РОЗДІЛ 5. ПРОГРАМНІ СИСТЕМИ ОБРОБКИ ТЕМПОРАЛЬНИХ МУЛЬТИМОДАЛЬНИХ ДАНИХ ЦИФРОВИХ ДВІЙНИКІВ ДОСЛІДЖУВАНИХ ОБ'ЄКТІВ**

Розроблений теоретичний інструментарій для обробки темпоральних мультимодальних даних та парадигма програмування мультиобразів дозволяють розроблювати на своїй основі програмні системи обробки темпоральних мультимодальних даних цифрових двійників. Розробимо узагальнену архітектуру програмної системи обробки темпоральних мультимодальних даних цифрових двійників та розглянемо варіанти її застосування для розроблення прикладного програмного забезпечення.

### **5.1. Архітектура програмної системи обробки темпоральних мультимодальних даних цифрових двійників**

Програмне забезпечення, що реалізує технологію цифрових двійників, має охоплювати всі аспекти процесів подання та обробки темпоральних мультимодальних даних цифрових двійників. Тому програмна система обробки темпоральних мультимодальних даних цифрових двійників має включати компоненти, які забезпечують:

- реалізацію правил та шаблонів синхронізації темпоральних даних;
- синхронізацію темпоральних мультимодальних даних;
- агрегацію темпоральних мультимодальних даних;
- отримання даних про досліджуваний об'єкт (фізичного двійника) від давачів та інших джерел даних;
- конвертацію форматів даних;
- реалізацію поведінкової моделі цифрового двійника;
- реалізацію візуальної моделі цифрового двійника;
- формування мультиобразу досліджуваного об'єкта;

- обробку даних цифрового двійника;
- аналіз даних поведінкової моделі цифрового двійника;
- відтворення даних візуальної моделі цифрового двійника;
- візуалізацію результатів дослідження фізичного двійника;
- забезпечення конфіденційності подання даних цифрового двійника;
- зберігання даних цифрового двійника.

Тоді узагальнена архітектура програмної системи обробки темпоральних мультимодальних даних цифрових двійників має вигляд, представлений на рис. 5.1.

Ця архітектура реалізує наступну логіку роботи з цифровим двійником. Досліджуваний об'єкт проявляє себе через низку фізичних процесів, явищ та сенсорних ефектів, які його характеризують та можуть бути зафіксовані у вигляді об'єктивних параметрів досліджуваного об'єкта за допомогою набору давачів та пристроїв реєстрації інформації, а також суб'єктивних параметрів, які оцінюються дослідником. Об'єктивні характеристики являють собою дані, що поділяються на поведінкові та мультимедійні.

Поведінкові дані у подальшому процесі дослідження об'єкта розглядаються та оброблюються як деякі технічні показники досліджуваного об'єкта. Мультимедійні дані використовуються для віртуальної візуалізації досліджуваного об'єкта та призначені для чуттєвого сприйняття людиною.

Всі дані, що отримуються в результаті реєстрації характеристик досліджуваного об'єкта, незалежно від їх типу, супроводжуються послідовністю часових міток, які визначають моменти часу отримання цих даних. Процедура отримання міток залежить від модальності даних, способу реєстрації цих даних та формату, в якому вони надходять до програмної системи обробки темпоральних мультимодальних даних цифрових двійників.



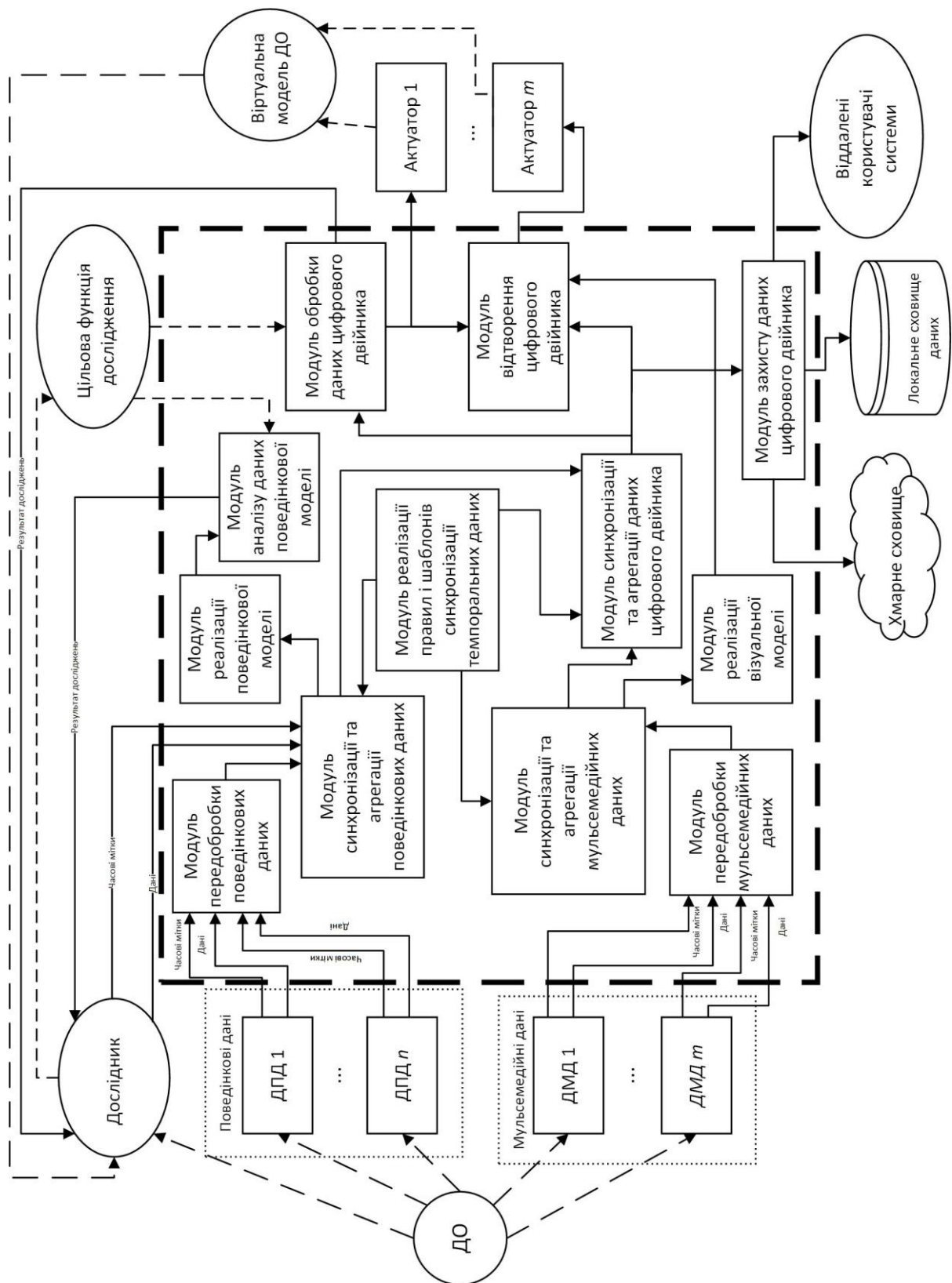


Рис. 5.1. Узагальнена архітектура програмної системи обробки темпоральних мультимодальних даних цифрових двійників

Поведінкові дані, разом із відповідними часовими мітками, надходять до модуля передобробки поведінкових даних, де відбувається їх перетворення з певного формату у формат внутрішнього подання, яке передбачає створення часткового мультиобразу для даних кожної модальності. Після конвертації форматів поведінкові дані надходять до модуля синхронізації та агрегації поведінкових даних.

Робота модуля синхронізації та агрегації поведінкових даних залежить від результату роботи модулю реалізації правил і шаблонів синхронізації темпоральних даних, в основу логіки якого покладені правила синхронізації даних. Додатково до архітектури програмної системи можуть бути включені компоненти, які забезпечують аналіз часових даних з метою автоматичного формування шаблонів синхронізації темпоральних мультимодальних даних.

Результатом роботи модуля синхронізації та агрегації поведінкових даних є мультиобраз, що відображає поведінку досліджуваного об'єкта. На основі цього мультиобразу модуль реалізації поведінкової моделі відтворює поведінку досліджуваного об'єкта.

Отримані з модулю реалізації поведінкової моделі дані результату моделювання поведінки досліджуваного об'єкта надходять до модулю аналізу поведінкових даних. Логіка роботи цього модуля визначається цільовою функцією, яку формулює дослідник перед початком дослідження. Результат аналізу поведінкових даних поведінкової моделі надається досліднику у певному заданому форматі та є одним з результатів роботи системи обробки темпоральних мультимодальних даних цифрових двійників.

Мульсемедійні дані, разом із відповідними часовими мітками, надходять до модуля передобробки мультсемедійних даних, де відбувається їх перетворення з певного формату у формат внутрішнього подання, яке передбачає створення часткового мультиобразу для даних кожної

модальності. Після конвертації форматів мультимедійні дані надходять до модуля синхронізації та агрегації мультимедійних даних.

Робота модуля синхронізації та агрегації мультимедійних даних залежить від результату роботи модулю реалізації правил і шаблонів синхронізації темпоральних даних. Результатом роботи модуля синхронізації та агрегації мультимедійних даних є мультиобраз, що відображає характеристики зовнішніх проявів досліджуваного об'єкта, в тому числі характеристики зовнішнього вигляду.

На основі цього мультиобразу модуль реалізації візуальної моделі відтворює зовнішній вигляд та інші зовнішні прояви досліджуваного об'єкта, в тому числі формує віртуальну модель досліджуваного об'єкта. Ця віртуальна модель є одним з результатів роботи програмної системи обробки темпоральних мультимодальних даних цифрових двійників.

Віртуальна модель передається для відтворення до модулю відтворення цифрового двійника. Додатково до архітектури програмної системи можуть бути включені компоненти, які забезпечують покращення якості візуалізації та застосування інших видів обробки даних візуальної моделі, зокрема виконують обробку графічних даних [7, 9, 10, 15, 17, 22, 24, 27, 28, 36, 47, 49, 65].

Мультиобрази, отримані в результаті роботи модуля синхронізації та агрегації поведінкових даних та модуля синхронізації та агрегації мультимедійних даних, надходять до модуля синхронізації та агрегації даних цифрового двійника, робота якого залежить від результату роботи модулю реалізації правил і шаблонів синхронізації темпоральних даних.

Результатом роботи модуля синхронізації та агрегації даних цифрового двійника є консолідовані дані цифрового двійника досліджуваного об'єкта. Ці дані передаються до модуля обробки даних цифрового двійника, логіка роботи якого визначається заданою цільовою

функцією дослідження. Результат обробки надається досліднику у деякій певній визначеній формі (форматі) та є одним з результатів роботи системи.

Крім того, консолідовані дані цифрового двійника надходять до модуля відтворення цифрового двійника, в якому відбувається корекція цифрового двійника за результатами обробки його даних у модулі обробки даних цифрового двійника та у відповідності до нових даних, що надходять від давачів, що реєструють мультимодальні параметри досліджуваного об'єкта.

Динамічно змінювані дані візуальної моделі цифрового двійника, що надходять до модуля відтворення цифрового двійника, подаються на набір актуаторів, кожен з яких дозволяє відтворювати дані певної модальності у вигляді певного аспекту образу цифрового двійника, тобто у вигляді віртуальної моделі досліджуваного об'єкта, яка демонструється досліднику, що є ще одним результатом роботи програмної системи обробки темпоральних мультимодальних даних цифрових двійників.

Дані цифрового двійника досліджуваного об'єкта надходять до модулю захисту даних цифрового двійника для шифрування та / або стеганографічного приховування конфіденційних даних цифрового двійника.

Після обробки конфіденційних даних дані цифрового двійника надходять для зберігання до локальних та / або хмарних сховищ, а також можуть бути надіслані віддаленим користувачам даних цифрового двійника досліджуваного об'єкта.

Розглянемо варіанти застосування цієї узагальненої архітектури для розроблення програмних систем обробки темпоральних мультимодальних даних цифрових двійників у галузі охорони здоров'я та освіти.

## **5.2. Програмне забезпечення медичних діагностичних систем на основі технології цифрових двійників**

Впровадження інформаційної технології цифрових двійників у галузі охорони здоров'я створює умови для розроблення систем підтримки прийняття медичних рішень на основі синхронізованих та консолідованих темпоральних мультимодальних медичних даних про стан здоров'я пацієнта, які надаватимуть лікарю інформацію про стан здоров'я пацієнта на якісно новому рівні. Система підтримки прийняття медичних рішень на основі технології цифрових двійників дозволить лікарю:

- аналізувати інформацію, подану з позицій часу;
- краще розуміти залежності між подіями лікування та результатами лікування (включаючи побічні ефекти для інших органів та систем);
- шукати подібні випадки, що трапились у пацієнта в минулому;
- моделювати можливий вплив нових схем лікування тощо.

Технологія цифрових двійників передбачає накопичення даних, які реєструються в результаті тривалого моніторингу реального об'єкта. Це відповідає випадку довготривалого моніторингу стану здоров'я пацієнта, особливо для пацієнтів із хронічними захворюваннями. Таким чином, ідея застосування цифрових двійників у галузі охорони здоров'я полягає у створенні та динамічному оновленні індивідуального набору даних пацієнта. Цей індивідуальний набір даних містить синхронізовані та агреговані мультимодальні дані, які отримуються в результаті різних медичних досліджень та подаються згідно з персоналізованою семантичною моделлю пацієнта, органів та систем органів його тіла. Ця персоналізована семантична модель має відображати хронічні захворювання та індивідуальні стани організму пацієнта. Тіло людини в цілому, а також окремі його органи та системи органів, можуть характеризуватися багатьма параметрами, виміряними за допомогою

широкого спектру медичного обладнання. Дані, які отримуються в результаті різних медичних досліджень та тестів, є мультимодальними, оскільки вони описують певний специфічний аспект функціонування людського організму. Для створення цифрового двійника фізичного двійника ці мультимодальні дані повинні бути синхронізовані та агреговані.

Мультимодальні дані для створення цифрового двійника пацієнта можуть бути отримані за допомогою широкого спектру пристроїв, інструментів та інструментів, які генерують мультимодальні дані у спеціалізованих, іноді унікальних форматах. Для адаптації узагальненої архітектури програмної системи обробки темпоральних мультимодальних даних цифрових двійників до специфіки медичної галузі, потрібно проаналізувати види медичного діагностичного обладнання та типи даних, що можуть бути отримані з його допомогою.

Усі медичні пристрої та інструменти можна розділити на декілька груп, включаючи діагностичне обладнання, обладнання медичної лабораторії, медичні монітори, терапевтичне обладнання, обладнання для життєзабезпечення тощо [348]. Діагностичні прилади та інструменти можна класифікувати як вимірювальний медичний інструмент (термометри, сфігмоманометри, глюкометри тощо), лабораторне обладнання (біохімічні аналізатори, гематологічні аналізатори тощо), медичне обладнання для візуалізації (апарати для ультразвукового дослідження, МРТ-апарати, ПЕТ-та КТ-сканери, апарати для рентгенографії тощо), функціональне діагностичне обладнання (електрокардіографи, електроенцефалографи, електроміографи, реографічні прилади, спірометри тощо). Ці пристрої та інструменти можуть бути цифровими або нецифровими. Вимірювання, отримані за допомогою нецифрових інструментів (наприклад, ртутного термометра, ручного сфігмоманометра), звичайно записуються вручну або як паперові, або як електронні записи. Вимірювання, отримані за допомогою цифрового обладнання, подаються у вигляді файлів певних

форматів залежно від типу використовуваного медичного обладнання. У деяких діагностичних процедурах, таких як ендоскопія, зокрема, колоноскопія, де результатом дослідження є потік відеоданих, використовуються формати файлів загального призначення. Найпоширеніші формати файлів медичних даних [349–356] представлені в табл. 5.1.

Таблиця 5.1 – Формати файлів медичних даних

Тип файлу даних	Розширення	Основний тип даних, що зберігається	Тип дослідження	Наявність часових міток
FEF	.FEF	Біологічний сигнал	Електрокардіограма	Є поле для послідовності часових значень
UFF	.BDF	Багатоканальний біологічний сигнал	Електроенцефалограма	Є чотири поля для значень часу: <ul style="list-style-type: none"> <li>– “Start date of recording”,</li> <li>– “Start time of recording”,</li> <li>– “Number of data records”,</li> <li>– “Duration of a data record, in seconds”</li> </ul>
LabPas HL7	.HL7	Результат аналізу крові	Аналіз крові	Є поле “Date/Time of the Observation”
UFF	.UFF	Дані УЗД	Ультразвукове дослідження	Є поле “local_time”
NIFTI	.NII	Дані MPT	Магнітно-резонансна томографія	Є зарезервоване поле для значень часу
DICOM	.DC3, .DCM, .DIC	Медичні зображення та метадані	Комп’ютерна томографія, рентгенографія тощо	Є поле “Study date” та “Study time”
CSV	.CSV	Цілі та дійсні числа (ASCII)	Спірометрія, термометрія, сфігмоманометрія тощо	Немає

Аналіз табл. 5.1 дозволяє зробити висновок про те, що більшість форматів медичних даних передбачає поле метаданих для зберігання часових міток, проте, структура файлів істотно відрізняється. Тому доцільно включити до архітектури програмного забезпечення медичної діагностичні системи на основі технології цифрових двійників модуль конвертації, який буде перетворювати файл кожного типу, що передбачається застосовувати у цій системі, у формат файлу-обгортки, розроблений у розділі 3. Компонентна діаграма [343, 344] цього модуля наведена на рис. 5.2.

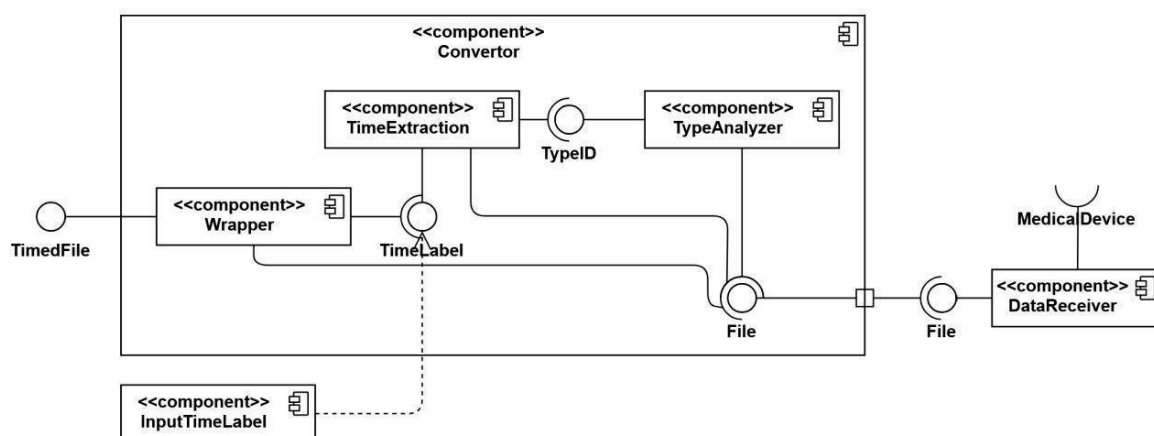


Рис. 5.2. Компонентна діаграма модуля конвертації медичних даних

Компонент *DataReceiver* призначений для отримання медичних даних з діагностичного обладнання. Компонент *TypeAnalyzer* призначений для аналізу типу вхідного файлу медичних даних. Компонент *TimeExtraction* призначений для отримання з файлу часових даних. Компонент *InputTimeLabel* призначений для введення часових даних людиною-оператором, що є альтернативним або допоміжним способом отримання часових даних програмною системою. Компонент *Wrapper* виконує формування файлу-обгортки, який є готовим до подальшого використання у модулі синхронізації та агрегації темпоральних даних.



Компонентна діаграма модуля синхронізації та агрегації медичних темпоральних мультимодальних даних показана на рис. 5.3.

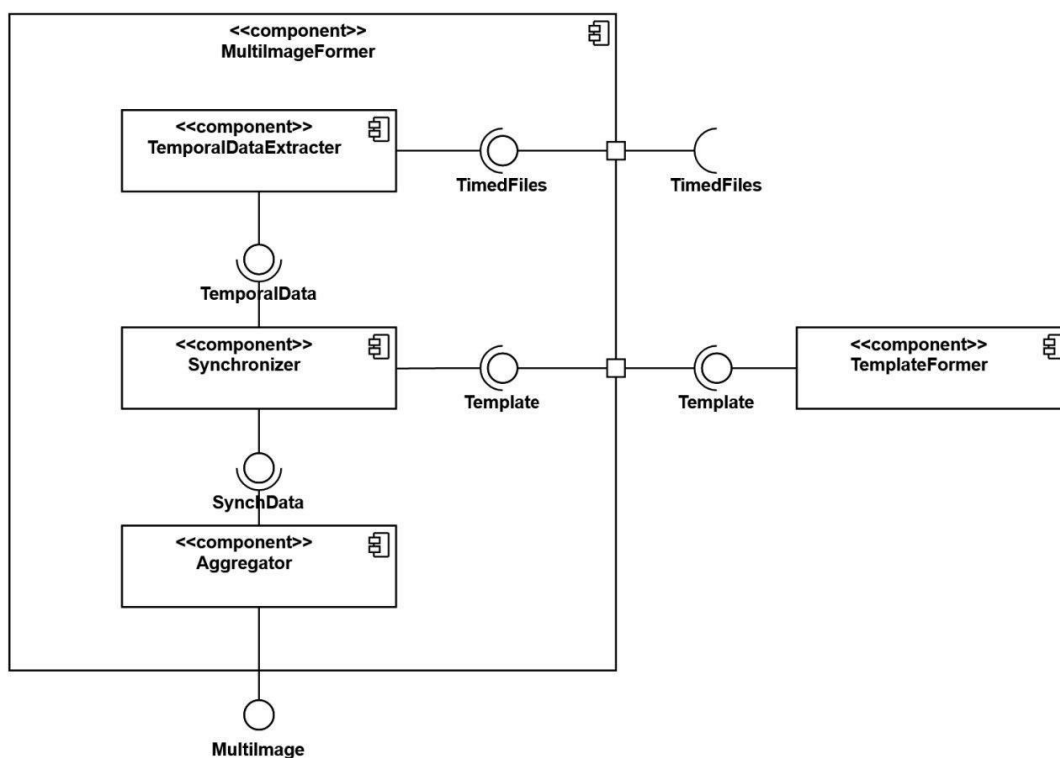


Рис 5.3. Компонентна діаграма модуля синхронізації та агрегації медичних темпоральних мультимодальних даних

Компонент *TemporalDataExtractor* призначений для перетворення даних з формату файлу-обгортки у внутрішнє подання темпоральних даних. Компонент *TemplateFormer* реалізує шаблони синхронізації, які використовуються компонентом *Synchronizer*. Компонент *Synchronizer* призначений для синхронізації темпоральних мультимодальних даних, отриманих з послідовності вхідних файлів, що подані у форматі файлу-обгортки. Компонент *Aggregator* призначений для агрегації синхронізованих темпоральних мультимодальних даних та формування мультимодального зображення, який готовий до подальшого використання для формування цифрового двійника.

Процес створення цифрового двійника пацієнта представлений на рис. 5.4. Перший етап цього процесу передбачає отримання даних про біомедичний об'єкт. Джерелом цих даних можуть бути медичні прилади, медичний інструментарій, лабораторне обладнання, медична документація. Дані можуть мати цифровий та нецифровий характер. Можливість отримання деяких медичних даних з нецифрових джерел, як-то, ртутний термометр, паперова медична картка пацієнта тощо, перетворює цю медичну діагностичну систему на ергатичну [357].

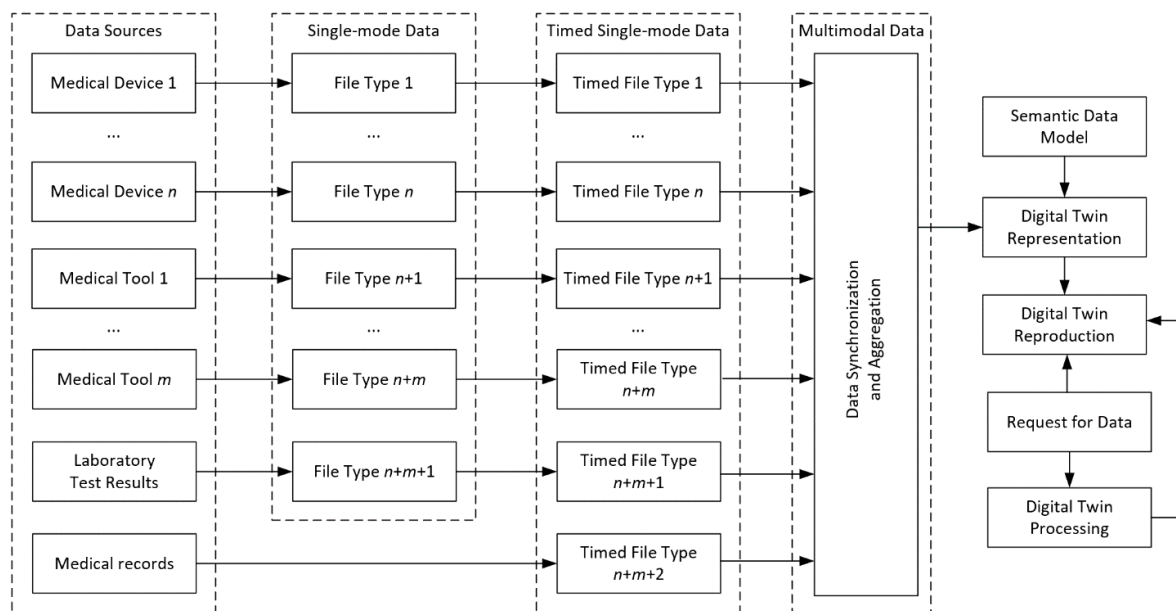


Рис. 5.4. Процес створення цифрового двійника пацієнта

Другим етапом процесу створення цифрового двійника пацієнта є перетворення кожного файлу в універсальний формат, який містить часові мітки у визначеному форматі файлу-обгортки. В результаті отримуємо файл з часовими мітками (*timed*-файл) для кожного оригінального файлу певного типу.

На третьому етапі відбувається синхронізація та агрегація даних, що подані у форматі файлу-обгортки. Процедура синхронізації використовує часові мітки, наявні у метаданих *timed*-файлів. Отриманий мультиобраз

надає комплексне подання даних цифрового двійника пацієнта.

Для створення цифрового двійника потрібно знати семантичну модель відповідного фізичного двійника, тобто пацієнта. Ця семантична модель повинна базуватися на таких медичних стандартах, як [358]. Персоніфікована семантична модель пацієнта – це вихідні дані для цієї інформаційної технології, а також дані, отримані з діагностичного медичного обладнання. Синхронізовані та агреговані дані зберігаються відповідно до семантичної моделі, яка створюється згідно з багаторівневою онтологічною моделлю, розробленою у розділі 3. Спосіб цільової обробки даних та варіанти відтворення цифрового двійника залежать від конкретних завдань моніторингу та дослідження стану здоров'я пацієнта. Для покращення якості візуалізації цифрового двійника передбачається використання додаткових етапів обробки візуальної моделі [7, 9, 36, 41].

Взаємодія модулів програмної системи, що реалізує процес створення цифрового двійника пацієнта, показана у вигляді діаграми послідовності [343, 344], що наведена на рис. 5.5.

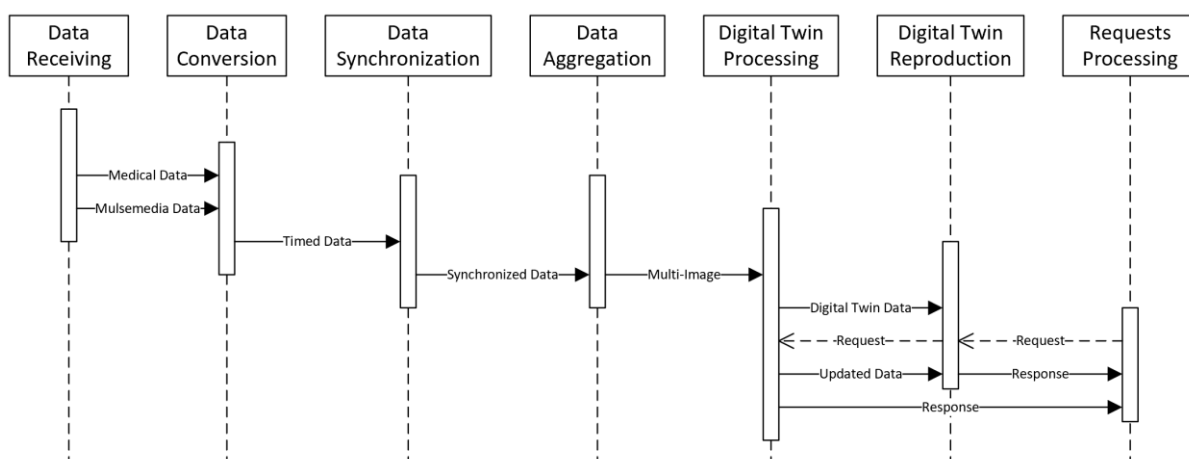


Рис. 5.5. Взаємодія модулів медичної програмної системи

Архітектура програмного забезпечення медичної діагностичної системи на основі технології цифрових двійників наведена на рис. 5.6.

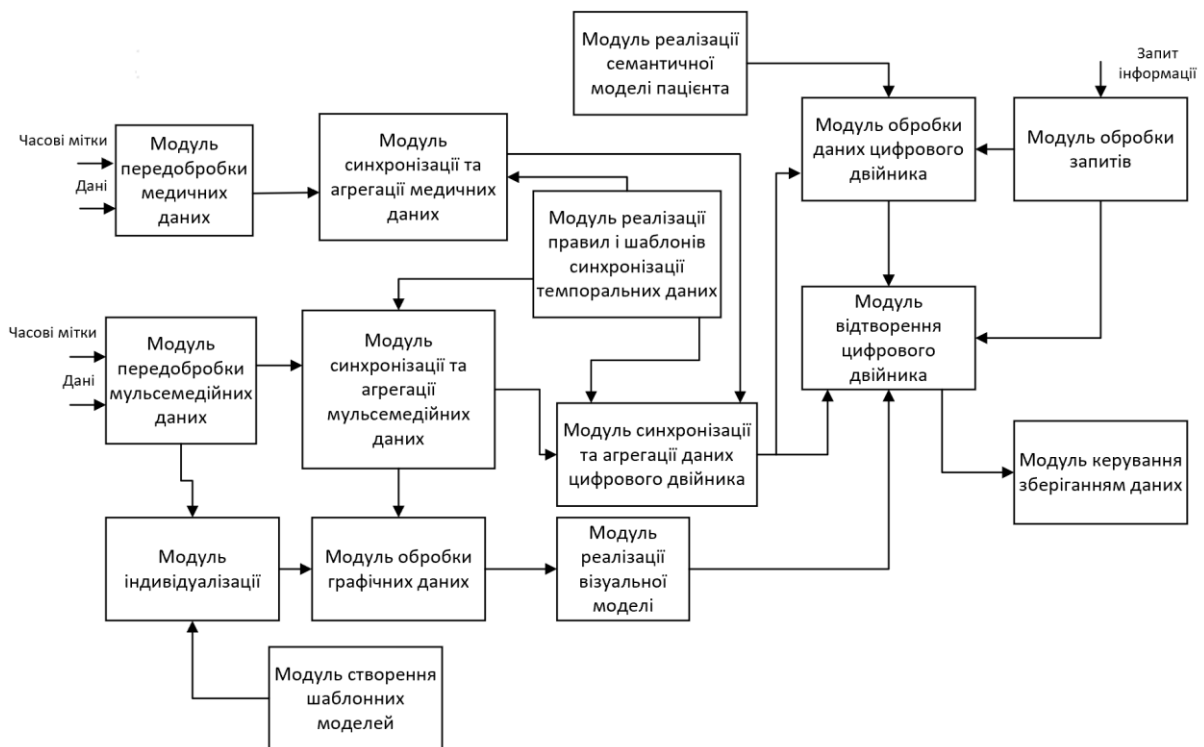


Рис. 5.6. Архітектура програмного забезпечення медичної діагностичної системи на основі технології цифрових двійників

Для спрощення візуального моделювання передбачається використання шаблонних моделей тіла людини, його частин та окремих органів. На основі вимірювань та медичних досліджень, проведених для конкретного пацієнта, шаблонна модель індивідуалізується шляхом зміни розмірів, пропорцій та інших показників, які впливають на візуальну модель. Додатково до архітектури медичної програмної системи можуть бути включені компоненти, які забезпечують застосування процедур покращення якості медичних графічних даних [14, 22, 24, 27, 52] та інших видів обробки даних візуальної моделі [7, 9, 36, 41].

Запропонована архітектура медичної програмної системи на основі технології цифрових двійників може бути застосована для розроблення експертних медичних систем з розширеними функціональними можливостями, програмного забезпечення систем e-Health [359] та m-Health [360], програмного забезпечення Hospital-at-Home [361] тощо.

### **5.3. Програмні системи дистанційного навчання на основі технології цифрових двійників та технології мультимедіа**

Розглянемо особливості розроблення програмного забезпечення з використанням технології цифрових двійників та технології мультимедіа для двох різновидів систем, що використовуються у галузі освіти: онлайн-лабораторії та імерсійного середовища. В обох випадках передбачається, що особи, що навчаються, досліджують (вивчають) певний реальний об'єкт (процес, явище тощо). Дослідження цього об'єкта відбувається шляхом реєстрації його характеристик за допомогою набору цифрових датчиків. Оскільки характеристики відображають різні сторони природи або поведінки об'єкта, отримуємо послідовності темпоральних мультимодальних даних, на основі яких формується та відтворюється цифровий двійник досліджуваного об'єкта.

#### **5.3.1. Архітектура онлайн-лабораторії**

Онлайн-лабораторія [34, 362–364] – це програмна система, яка дозволяє учням використовувати віддалене обладнання для проведення експериментів для навчальних цілей. Програмне забезпечення онлайн-лабораторії включає програмні засоби для обробки, аналізу, зберігання, передачі даних та візуалізації результатів.

Архітектуру програмного забезпечення онлайн-лабораторії, що ґрунтується на використанні технології цифрових двійників та технології мультимедіа, наведено на рис. 5.7.

Послідовності темпоральних даних двох категорій (поведінкові дані та мультимедійні дані) надходять до програмної системи у режимі реального часу з  $N$  цифрових датчиків, з якими безпосередньо взаємодіють модуль реєстрації поведінкових даних та модуль реєстрації мультимедійних

даних. Причому,  $N = N_1 + N_2$ , де  $N_1$  – кількість давачів, що реєструють поведінкові дані та  $N_2$  – кількість давачів, що реєструють мультимедійні дані. В результаті вимірювання характеристик досліджуваного об'єкта утворюються  $N$  пар послідовностей  $(\bar{t}^k, \bar{s}^k)$ , таких, що  $\bar{t}^k = \langle t_i^k \rangle_{i=1}^{n_k}$ ,  $t_i^k \in T$ ,  $\bar{s}^k = \langle s_i^k \rangle_{i=1}^{n_k}$ ,  $s_i^k \in M_k$ ,  $t_{n_k}^k \leq L_{exp}$ , де  $\bar{t}^k$  – це послідовність значень, що визначають моменти часу  $t_i^k$ , коли вимірюються значення  $s_i^k$ , які відображають характеристику об'єкта,  $T$  – множина часових значень,  $M_k$  – множина даних  $k$ -ї модальності,  $L_{exp}$  – тривалість експерименту. Ці темпоральні мультимодальні дані, залежно від їх категорії, надходять до модулю синхронізації та агрегації поведінкових даних або до модулю синхронізації та агрегації мультимедійних даних.

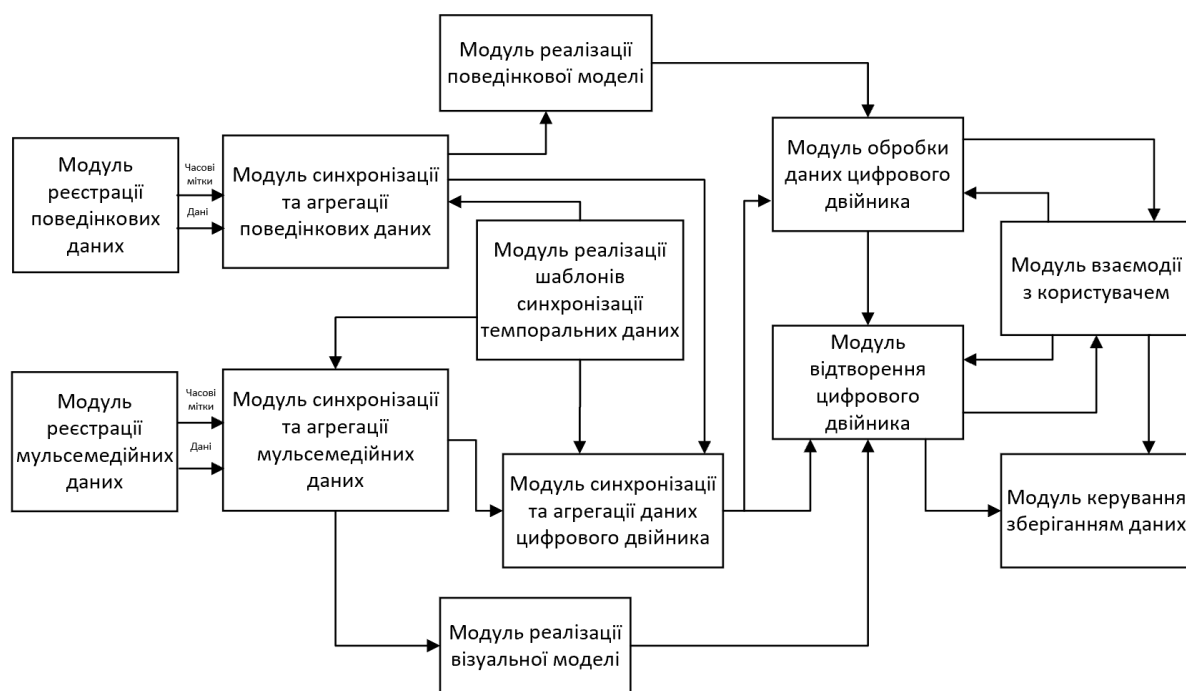


Рис. 5.7. Архітектура програмного забезпечення онлайн-лабораторії

Синхронізація має особливе значення, коли віддалений зв'язок із давачами є нестабільним через високе навантаження мережі або інші

зовнішні умови. Синхронізація відбувається згідно з шаблонами синхронізації, що визначені відповідно до режимів роботи давачів даних.

Синхронізовані та агреговані дані обох категорій надходять до модуля синхронізації та агрегації даних цифрового двійника, а також дані кожної категорії надходять до модуля реалізації моделі відповідного типу (поведінкової або візуальної). Результати моделювання поведінки об'єкта надходять до модуля обробки даних цифрового двійника. Результати моделювання зовнішнього вигляду та інших зовнішніх проявів об'єкта надходять до модуля відтворення цифрового двійника. Спосіб відтворення цифрового двійника залежить від технічних можливостей на стороні користувача онлайн-лабораторії. В разі відсутності у користувача обладнання для відтворення мультимедійних ефектів візуалізація відбувається лише у графічний спосіб, тобто відтворюються візуальні або аудіовізуальні дані, а дані інших мультимедійних типів відтворюються у вигляді графіків, таблиць, текстових описів тощо. Безпосередня комунікація користувача з системою відбувається через модуль взаємодії з користувачем, який може бути реалізований, наприклад, як веб-застосунок.

Модуль обробки даних цифрового двійника реалізує алгоритми обробки даних відповідно до мети онлайн-лабораторії, характеру експериментів, наявного обладнання тощо. Вхідними даними цього модуля є консолідовані дані цифрового двійника та дані, що отримуються в результаті моделювання поведінки досліджуваного об'єкта, а також запити, що отримуються від користувача системи у процесі його роботи (експериментів) з онлайн-лабораторією. Для збереження результатів експериментів та інших даних користувача використовується локальна база даних; для збереження даних великого обсягу, що оброблюються у онлайн-лабораторії, передбачається використання сховища даних.

Діаграма розгортання [343, 344] основних компонентів онлайн-лабораторії наведена на рис. 5.8.

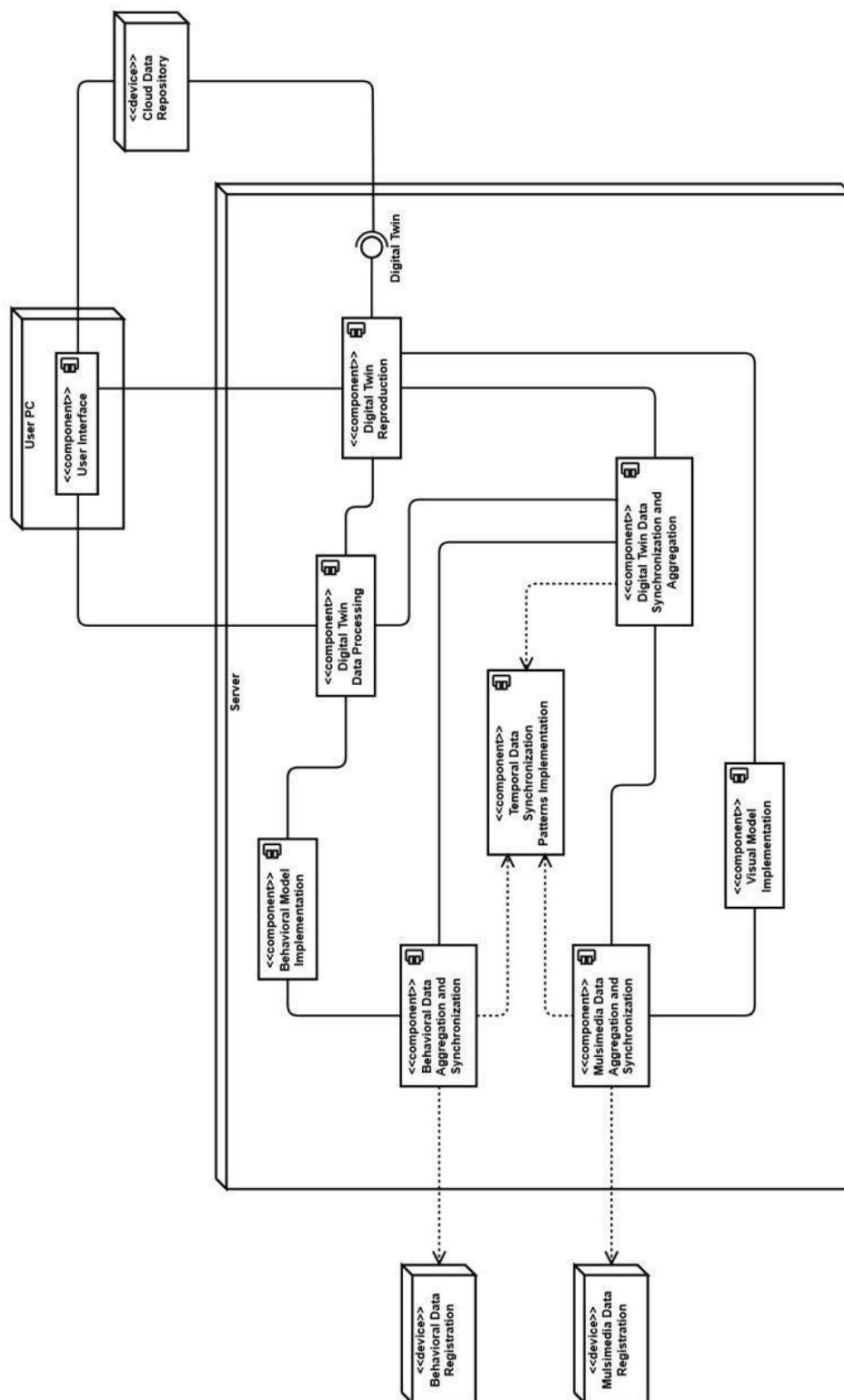


Рис. 5.8. Діаграма розгортання програмного забезпечення онлайн-лабораторії



Як показано на діаграмі розгортання, компоненти програмного забезпечення онлайн-лабораторії встановлюються на вузли п'ятих типів. Програмне забезпечення для реєстрації поведінкових даних та мультимедійних даних встановлюється на комп'ютерне обладнання системи давачів. Програмний застосунок, що призначений для взаємодії користувача з онлайн-лабораторією встановлюється на пристрій користувача (ноутбук, планшет, смартфон тощо). Програмне забезпечення, що реалізує решту функцій системи, встановлюється на сервер.

Як було зазначено вище, онлайн-лабораторія передбачає, але не вимагає відтворення мультимедійних даних на стороні користувача. Цим онлайн-лабораторія відрізняється від імерсійного середовища, де відтворення мультимедійних даних є обов'язковим. Розробимо архітектуру імерсійного середовища, що ґрунтується на технології цифрових двійників.

### **5.3.2. Архітектура імерсійного середовища**

Імерсійне середовище [33, 37, 38, 365–367] – це апаратно-програмна система, яка дозволяє імітувати присутність користувача у штучно створеній сцені (віртуальному світі), де користувач може взаємодіяти у віртуальний спосіб з реальними та віртуальними об'єктами, процесами або явищами.

Апаратна частина імерсійного середовища складається з цифрових давачів, які відстежують дії користувача в імерсійному середовищі, та актуаторів, що роблять можливим тривимірну візуалізацію, просторове відтворення звуку, тактильну взаємодію користувача з віртуальними об'єктами тощо. Програмна частина імерсійного середовища включає програмні засоби для візуалізації віртуальних об'єктів та віртуальної взаємодії з реальними об'єктами. Архітектура програмного забезпечення імерсійного середовища наведена на рис. 5.9.

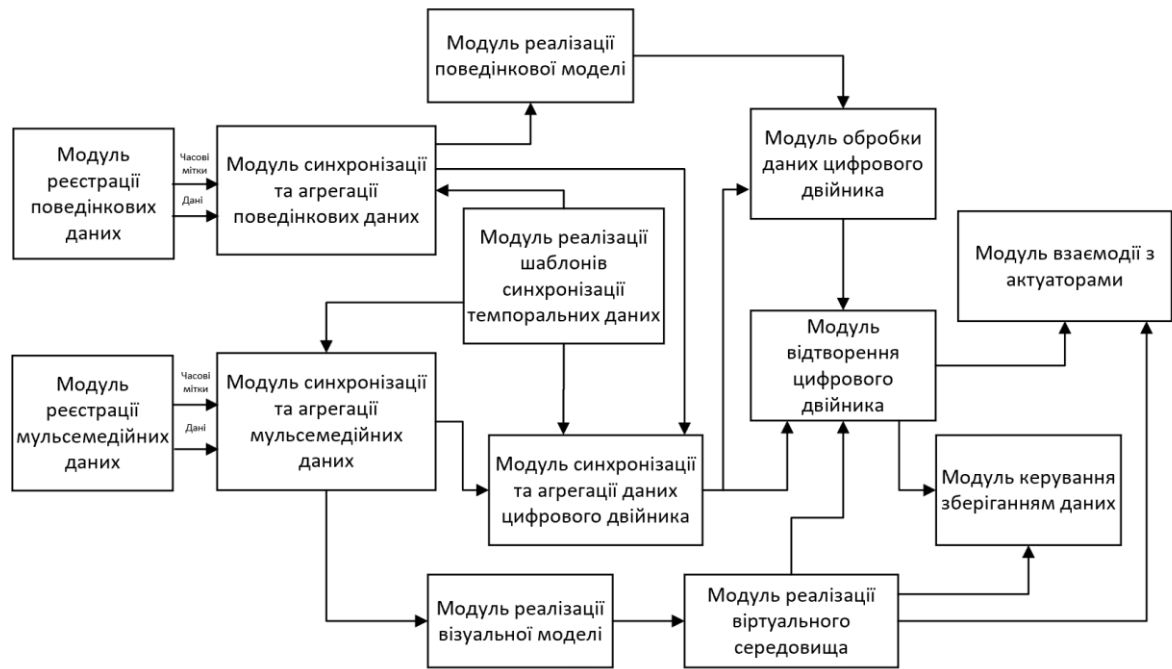


Рис. 5.9. Архітектура програмного забезпечення імерсійного середовища

Основними даними для програмної системи, що реалізує імерсійне середовище на основі технології цифрових двійників, є наступні дані:

- поведінкові дані фізичного двійника;
- мультимедійні дані фізичного двійника;
- дані цифрового двійника;
- дані віртуальної модулі сцени.

Поведінкові та мультимедійні дані фізичного двійника надходять для синхронізації та агрегації до відповідного модуля синхронізації та агрегації, де відбувається їх обробка відповідно до шаблонів синхронізації, від яких залежить від типу давачів, що застосовуються для моніторингу фізичного двійника.

Консолідовані дані кожної категорії (поведінкові дані, мультимедійні дані) надходять до модулю реалізації моделі відповідного виду, а також до модулю синхронізації та агрегації даних цифрового двійника. Сформована візуальна модель цифрового двійника надається для обробки до модуля реалізації віртуального середовища, яке формує вигляд віртуальної сцени.

Дані про сформовану віртуальну сцену надходять до модуля відтворення цифрового двійника та подаються на модуль взаємодії з актуаторами.

Результати моделювання поведінки фізичного двійника та консолідовані дані цифрового двійника передаються з модуля реалізації поведінкової моделі на модуль обробки даних цифрового двійника, де відбувається цільова обробка цих даних, яка залежить від призначення імерсійного середовища. Результати цієї обробки надходять до модуля відтворення цифрового двійника для уточнення та динамічної зміни цифрового двійника.

Оскільки дані цифрового двійника та віртуального середовища мають великий об'єм, передбачається їх зберігання у сховищах даних; керування процесом зберігання цих даних забезпечує відповідний модуль.

Якщо передбачається активна взаємодія користувача з елементами віртуального простору, то додатково до архітектури імерсійної системи мають бути включені модуль реєстрації дій користувача, який передбачає взаємодію з відповідними давачами даних, та модуль зворотного зв'язку, який генерує реакцію системи на дії користувача.

Діаграма розгортання [343, 344] основних компонентів імерсійного середовища наведена на рис. 5.10.

Як показано на діаграмі розгортання, компоненти програмного забезпечення імерсійного середовища встановлюються на вузли п'ятих типів. Програмне забезпечення для реєстрації поведінкових даних встановлюється на комп'ютерне обладнання системи давачів, які реєструють технічні характеристики фізичного двійника. Програмне забезпечення для реєстрації мультимедійних даних встановлюється на комп'ютерне обладнання системи давачів, які реєструють зовнішні прояви фізичного двійника. Програмне забезпечення, що реалізує решту функцій системи, встановлюється на сервер. Окремими вузлами є вузол сховища даних та вузол системи актуаторів.

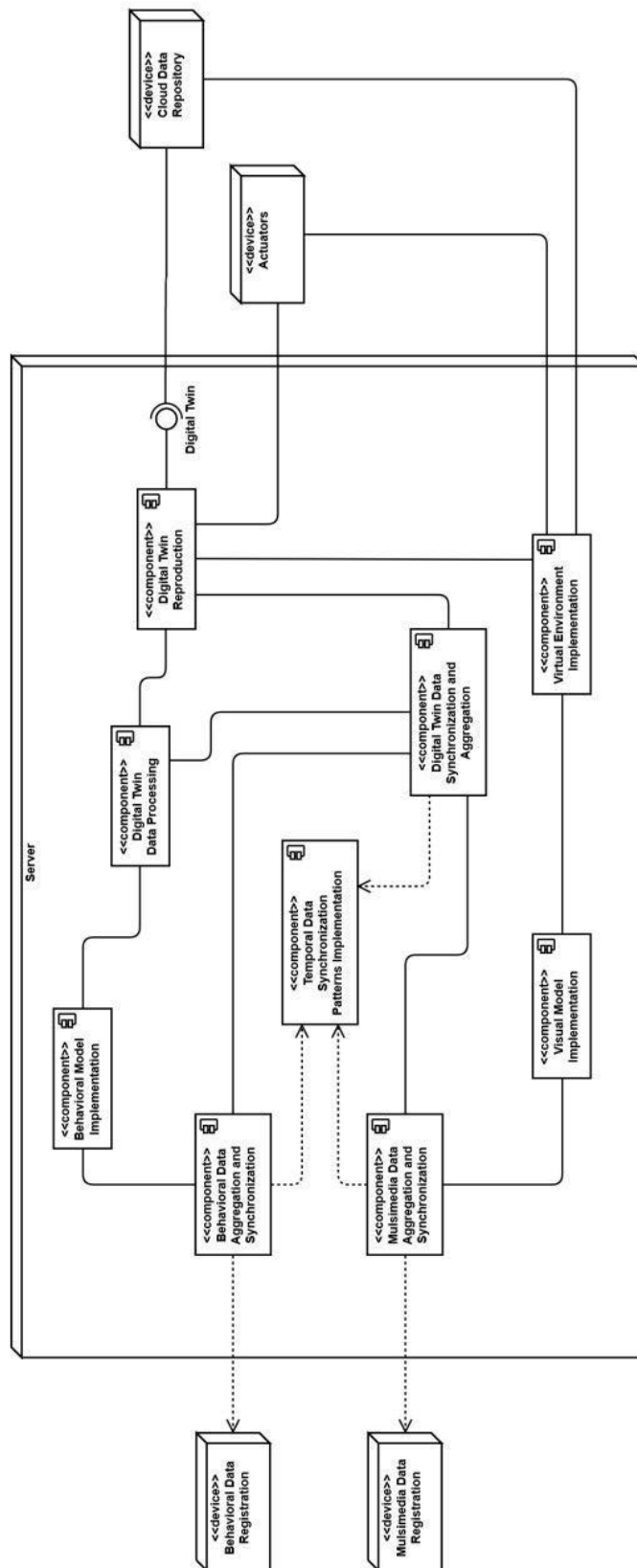


Рис. 5.10. Діаграма розгортання програмного забезпечення імерсійного середовища

Одним з варіантів імерсійного середовища є імерсійна телеконференц-система, яка призначена для реєстрації, передавання та відтворення не лише аудіовізуальної інформації, а й інформації про навколишнє середовище, де присутні користувачі. Метою застосування імерсійної телеконференц-системи створення у її учасників ефекту присутності, що досягається за рахунок реєстрації, передачі та відтворення мультимодальної інформації про всю сцену, де присутні учасники телеконференції [38]. Основною вимогою до імерсійної телеконференц-системи є забезпечення достатньо високого рівня QoE та QoS. Це може бути досягнуто за рахунок використання технології WebRTC [35].

Таким чином, розроблена узагальнена архітектура програмної системи обробки темпоральних мультимодальних даних цифрових двійників може бути адаптована для розроблення різноманітного програмного забезпечення, що ґрунтується на технології цифрових двійників, зокрема для розроблення програмних систем обробки темпоральних мультимодальних даних для галузі охорони здоров'я і освіти.

## **5.4. Висновки до розділу 5**

У цьому розділі вирішувалась науково-практична задача розроблення архітектури програмних систем обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів та моделювання програмного забезпечення прикладних програмних систем на її основі. В результаті отримано наступні результати.

1. Розроблено узагальнену архітектуру програмної системи обробки темпоральних мультимодальних даних цифрових двійників, яка дозволяє виконувати повний цикл дослідження об'єкта. Розроблена архітектура забезпечує динамічне збирання двох типів характеристик досліджуваного об'єкта: поведінкових даних та мультимедійних даних, які

використовуються для створення складових цифрового двійника: поведінкової моделі та візуальної моделі. В результаті консолідації даних цих моделей відбувається формування цифрового двійника досліджуваного об'єкта. Сформований цифровий двійник підлягає цільовій обробці, логіка якої визначається метою дослідження та призначенням програмної системи обробки темпоральних мультимодальних даних цифрових двійників.

2. Розроблено архітектуру програмного забезпечення медичної діагностичної системи на основі технології цифрових двійників, яка конкретизує узагальнену архітектуру програмної системи обробки темпоральних мультимодальних даних цифрових двійників. Виконане модулювання окремих компонентів, які реалізують функціональність відповідних модулів розробленої архітектури програмного забезпечення медичної діагностичної системи. Визначено процес створення цифрового двійника пацієнта, який ґрунтується на використанні персоніфікованої семантичної моделі пацієнта та індивідуалізації шаблонних моделей тіла людини, його частин та окремих органів. Запропонована медична діагностична система не має аналогів, оскільки ґрунтується на застосуванні технології цифрових двійників.

3. Розроблено архітектуру програмного забезпечення онлайн-лабораторії на основі технології цифрових двійників та технології мультимедіа, яка призначена для дистанційного проведення експериментів над досліджуваним об'єктом. Запропонована архітектура онлайн-лабораторії ґрунтується на узагальненій архітектурі програмної системи обробки темпоральних мультимодальних даних цифрових двійників. Додатково до архітектури програмної системи включено компоненти, які реалізують процедуру віддаленого доступу користувача до даних цифрового двійника досліджуваного об'єкта. Промодельовано розгортання програмної системи онлайн-лабораторії з врахуванням специфіки апаратного забезпечення процесу дослідження об'єкта. Запропонована

онлайн-лабораторія не має аналогів, оскільки ґрунтується на застосуванні технології цифрових двійників.

4. Розроблено архітектуру програмного забезпечення імерсійного середовища на основі технології цифрових двійників, яке призначене для віртуальної взаємодії користувача з досліджуваним об'єктом із забезпеченням ефекту присутності. Запропонована архітектура програмної системи імерсійного середовища є варіантом реалізації узагальненої архітектури програмної системи обробки темпоральних мультимодальних даних цифрових двійників. Додатково до архітектури програмної системи включено компоненти, які реалізують створення віртуального середовища та його взаємодію з цифровим двійником досліджуваного об'єкта. Промодельовано розгортання програмної системи імерсійного середовища з врахуванням специфіки апаратного забезпечення процесу дослідження фізичного двійника та імерсійного відтворення цифрового двійника досліджуваного об'єкта. Запропоноване імерсійне середовище не має аналогів, оскільки ґрунтується на застосуванні технології цифрових двійників.

## ВИСНОВКИ

У дисертаційній роботі вирішено актуальну науково-технічну проблему – підвищення ефективності обробки темпоральних мультимодальних даних в комп'ютерних системах.

У рамках виконаних наукових досліджень дістали подальший розвиток основи побудови програмного забезпечення систем обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів. При цьому отримано такі основні наукові й практичні результати.

1. Показано, що для формальної специфікації моделей цифрового двійника досліджуваного об'єкта доцільно подавати темпоральні мультимодальні дані, які характеризують об'єкт, у вигляді мультиобразу – впорядкованої сукупності синхронізованих даних. Уперше розроблено метод створення мультиобразу досліджуваного об'єкта, який полягає в послідовному виконанні процедур: формування структури даних, декомпозиція мультиобразу на набір часткових мультиобразів, підготовка темпоральних мультимодальних даних, об'єднання, сортування та проріджування даних, що надає можливість створювати універсальне програмне забезпечення систем обробки даних цифрових двійників досліджуваних об'єктів.

2. Доведено, що для синхронізації мультиобразів, які потребують консолідованої обробки, доцільно використовувати правила синхронізації: універсальне правило, базові правила, правила нечіткої синхронізації. Уперше розроблено метод синхронізації мультиобразів, який ґрунтується на застосуванні інтервальних відношень, виконанні правил синхронізації, а також процедур: формування шаблону синхронізації, впорядкування мультиобразів, попарна синхронізація мультиобразів, що робить можливим комплексне дослідження кількох об'єктів.

3. Для опису досліджуваного об'єкта уперше розроблено алгебраїчну систему, що задається трьома множинами – множиною елементів (агрегатів), множиною операцій та множиною відношень. Показано, що визначальними рисами розробленої алгебраїчної системи є: оперування



агрегатами – складеними математичними об'єктами, компоненти яких є впорядкованими, врахування порядку слідування компонентів при виконанні операцій, використання операцій впорядкування та спеціальних відношень (частотних, інтервальних) між компонентами агрегатів, що дає змогу спростити алгоритмічне забезпечення процесів обробки темпоральних мультимодальних даних (в тому числі нечітко визначених) завдяки використанню уніфікованих процедур обробки даних.

4. Уперше розроблено моделі цифрового двійника: муксельну модель, часову зв'язну модель та багаторівневу онтологічну модель, які дозволяють представляти досліджуваний об'єкт на різних рівнях абстракції. Муксельна модель призначена для подання деталізованої інформації про об'єкт дослідження та, на відміну від відомих, представляє об'єкт як композицію мукселів – об'ємних елементів, які описуються впорядкованою сукупністю темпоральних мультимодальних даних. Часова зв'язна модель слугує для відображення об'єкта в динаміці зміни його стану та, на відміну від відомих, ґрунтується на використанні часових ключів для формування впорядкованої послідовності станів об'єкта та визначенні зв'язків з джерелами даних для динамічної реєстрації інформації про досліджуваний об'єкт. Багаторівнева онтологічна модель, на відміну від відомих, подає досліджуваний об'єкт як композицію компонентів, кожен з яких характеризується мультисемедійними властивостями та описується поведінковими характеристиками, які визначаються темпоральними мультимодальними даними, отриманими від досліджуваного об'єкта, та може включати до свого складу муксельну та часову зв'язну моделі для деталізованого визначення окремих компонентів об'єкта, що дає змогу описувати досліджуваний об'єкт максимально повно.

5. Уперше запропоновано парадигму програмування для реалізації ефективних обчислень у комп'ютерних системах, яка, на відміну від існуючих, ґрунтується на використанні нової сутності – мультиобразу досліджуваного об'єкта як основного компонента обчислювального процесу, що уможливорює спрощення процесу розроблення програмного

забезпечення систем обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів. Показано, що особливістю цієї парадигми є її зорієнтованість на застосування апарату алгебраїчної системи агрегатів для обробки темпоральних мультимодальних даних, використання синхронізації та агрегації даних різних модальностей як основної складової моделі обчислень, застосування різнорідних джерел даних. Реалізація парадигми програмування дозволяє підвищити ефективність процесу розроблення прикладного програмного забезпечення систем на основі технології цифрових двійників за рахунок зниження складності програмного коду та поліпшення зручності його супроводження.

6. Розроблено спеціалізовану мову програмування ASAMPL, яка реалізує запропоновану парадигму програмування та призначена для розроблення прикладного програмного забезпечення систем обробки темпоральних мультимодальних даних, зокрема мультимедійних та мультисемедійних застосунків. Обробка мультимодальних даних ґрунтується на використанні спеціальних структур даних – мультиобразів, які реалізуються за допомогою агрегатів та кортежів. Доведено, що застосування спеціальних операторів обробки темпоральних мультимодальних даних (часовий оператор, оператор заміни, оператори вивантаження та завантаження, оператор відтворення) дозволяє зменшити обсяг програмного коду в 4-6 разів, зменшити цикломатичну складність програми у 2-3 рази та підвищити індекс зручності підтримки програмного коду на 25-30% порівняно з мовами програмування C#, C++, C та Python.

7. Розроблено архітектуру програмної системи для обробки темпоральних мультимодальних даних цифрових двійників та запропоновано її варіанти для галузі охорони здоров'я та для дистанційної освіти. Показано, що особливістю розробленої архітектури є реалізація дворівневої процедури синхронізації та агрегації темпоральних мультимодальних даних, що дозволяє спростити процес розроблення прикладного програмного забезпечення систем обробки темпоральних мультимодальних даних цифрових двійників досліджуваних об'єктів.

## СПИСОК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Mathematical Methods in Interdisciplinary Sciences / Sulema Ye., Kerre E., et al.; editor Chakraverty S. Wiley, USA, 2020. 464 p. ISBN : 9781119585640. DOI : 10.1002/9781119585640.
2. Sulema Ye.S., Rvach D.V. Models of computation for Digital Twins data processing. *Наукові вісти КПІ*, 2020. № 2, С. 74–81.
3. Sulema Ye. Multimodal data processing based on algebraic system of aggregates relations. *Radio Electronics, Computer Science, Control*, 2020. № 1, С. 169–180.
4. Sulema Ye., Peschanskii V. Timewise data processing with programming language ASAMPL. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Технічні науки*, 2020. Т.31(70), Част.1. № 1, С. 132–137.
5. Дичка І.А., Сулема Є.С. Модель подання мультимодальних даних для комплексного опису об'єктів спостереження. *Вісник Вінницького політехнічного інституту*, 2020. № 1, С. 53–60.
6. Sulema Ye., Glinskii V. Semantics and pragmatics of programming language ASAMPL. *Проблеми програмування*, 2020. № 1, С. 74–83.
7. Sulema Ye., Rudenko C. A method of artifact compensation for dual quaternion skinning and its application in digital twin models. *Вісник Хмельницького національного університету. Технічні науки*, 2020. № 1, С. 12–19.
8. Сулема Є.С., Радченко Є.О. Метод стеганографічного захисту мультимедійних даних на основі процедури псевдовипадкового вбудовування. *Наукові вісти КПІ*, 2020. № 1, С. 40–47.
9. Sulema Y.S., Los I.A. Levels-Of-Detail generation method for skeletal meshes. *Системні технології*, 2019. Т.6. № 125, С. 3–14.

- 10.Сулема Є.С., Топчієв Б.С. Інтелектуальна колоризація зображень за допомогою генеративних змагальних мереж. *Системні технології*, 2019. Т.5. № 124. С. 94–103.
- 11.Dychka I., Sulema Ye., Bukhtiarov Iu. Digital Twin Information Technology for Biomedical Data Complex Representation and Processing. *Вісник Херсонського національного технічного університету*, 2019. № 3 (70), С. 112–119.
- 12.Dychka I.A., Sulema Ye.S. Ordering Operations in Algebraic System of Aggregates for Multi-Image Data Processing. *Наукові вісті КНУ*, 2019. № 1, С. 15–23.
- 13.Dychka I.A., Sulema Ye.S. Logical Operations in Algebraic System of Aggregates for Multimodal Data Representation and Processing. *Наукові вісті КНУ*, 2018. № 6, С. 44–52.
- 14.Shkurat O.S., Sulema Ye.S., Dychka A.I. Complicated Shapes Estimation Method for Objects Analysis in Video Surveillance Systems. *Наукові вісті КНУ*, 2018. № 3, С. 53–62. DOI : 10.20535/1810-0546.2018.3.136433.
- 15.Dychka I.A., Sulema Ye.S., Chernykh D.A. Rasterization Method for Voxel Model Cutting. *Наукові вісті КНУ*, 2018. № 2, С. 25–32.
- 16.Dychka I., Shyrochyn S., Sulema Ye. Analysis of Parallel Computations Efficiency for User's Private Multimedia Data Protection in Clouds. *Наукові вісті КНУ*, 2016. Т.1, С. 40–46.
- 17.Резнік К.Ю., Сулема Є.С. Метод локалізації вушної раковини на зображенні людини у профіль. *Вісник Хмельницького національного університету*, 2014. № 4, С. 113–119.
- 18.Сулема Є.С., Широчин С.С. Спосіб стеганографічного захисту даних в аудіо-файлах на основі комплементарного образу. *Наукові вісті КНУ*, 2014. Т.61, С. 80–87.
- 19.Сулема Є.С., Широчин С.С. Спосіб стеганографії зображень на основі комплементарного образу. *Захист інформації*, 2013. Т.4, С. 345–353.

- 20.Сулема Є.С., Широчин С.С. Спосіб стеганографії зображень з фрагментацією стегоданих та розділенням закритого ключа. *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні*, 2012. Т.1(22), С. 64–68.
- 21.Sulema Ye. ASAMPL: Programming Language for Mulsemedia Data Processing Based on Algebraic System of Aggregates. *Advances in Intelligent Systems and Computing*, 2018. Vol.725, P. 431–442. ISSN : 21945357.
- 22.Shkurat O., Sulema Ye., Suschuk-Sliusarenko V., Dychka A. Image Segmentation Method Based on Statistical Parameters of Homogeneous Data Set. *Advances in Intelligent Systems and Computing*, 2019. Vol.902, P. 271–281. ISSN : 21945357.
- 23.Radchenko Ye., Dychka I., Sulema Ye., Suschuk-Sliusarenko V., Shkurat O. Steganographic Protection Method Based on Huffman Tree. *Advances in Intelligent Systems and Computing*, 2019. Vol.902, P. 283–292. ISSN : 21945357.
- 24.Hu Zh., Dychka I., Sulema Ye., Valchuk Yu., Shkurat O. Method of medical images similarity estimation based on feature analysis. *International Journal of Intelligent Systems and Applications (IJISA)*, 2018. Vol.10. No. 5, P. 14–22. ISSN : 20749058.
- 25.Hu Zh., Dychka I., Sulema Ye., Radchenko Ye. Graphical Data Steganographic Protection Method Based on Bits Correspondence Scheme. *International Journal of Intelligent Systems and Applications (IJISA)*, 2017. Vol.9. No.8, P.34–40. ISSN : 20749058.
- 26.Sulema Ye., Dychka I., Sulema O. Multimodal Data Representation Models for Virtual, Remote, and Mixed Laboratories Development. *Lecture Notes in Networks and Systems*, 2018. Vol.47, P. 559–569. ISSN : 23673389.
- 27.Yevgeniya Sulema, Etienne Kerre, Oksana Shkurat. Vector Image Retrieval Methods Based on Fuzzy Patterns. *International Journal of Modern*

- Education and Computer Science (IJMECS)*. 2020. Vol.12. No.3. P. 8–16.  
DOI : 10.5815/ijmecs.2020.03.02.
28. Sulema Ye., Papusha R. Method of Raster Line Dynamic Smoothing. *Геометричне моделювання та інформаційні технології*, 2017. № 2 (4), С. 84–89.
  29. Сулема Є.С., Широчин С.С. Метод захисту зображень на основі шифрування палітри. *Вісник Хмельницького національного університету*, 2014. №3, С. 114–119.
  30. Сулема Е.С., Ебрахими Кахоу С. Адаптивный способ сжатия изображений. *Вісник Хмельницького національного університету*, 2010. № 2, С. 125–131.
  31. Павловець О.В., Сулема Є.С. Метод розпізнавання мовних одиниць на основі ADABOOST. *Вісник Хмельницького національного університету*, 2013. № 4, С. 211–222.
  32. Sulema Ye., Ebrahimi Kahou S. Comparative Study of Image Compression Algorithms. *Radioelectronics and Informatics*, 2009. № 1. P. 1–8.
  33. Сулема Є.С., Ебрахімі Кахоу С., Лавренчук С.С. Віртуальна мультимедійна лабораторія: особливості організації та обробки даних. *Збірник наукових праць Національного гірничого університету*, 2009. Vol.2. № 33. С. 90–96.
  34. Sulema Ye., Cvjetkovic V. Remote Laboratory for Supporting e-Studies in Electronics. *International Journal of Online Engineering*, 2006. Vol 2, № 1, P. 1–9.
  35. Sulema Ye., Amram N., Aleshchenko O., Sivak O. Quality of Experience Estimation for WebRTC-based Video Streaming. Proceedings of the 24th International Conference “European Wireless” (EW2018). Catania, Italy, 2018. P. 1–6.
  36. Dychka I., Sulema Ye., Rudenko C. A Mathematical Model of Microsurface Normal Distribution for Specular Bidirectional Reflectance Distribution

- Function. Proceedings of the International Conference “Advanced Computer Information Technologies” (ACIT 2018). Ceske Budejovice, Czechia, 2018, P. 30–33.
37. Sulema Ye., Rozinaj G. WebRTC-based 3D Videoconferencing System. Proceedings of 59th International Symposium ELMAR-2017. Zadar, Croatia, 2017, P. 193–196.
38. Sulema Ye., Rozinaj G. Multimodal Information in Telecommunication Systems: Analysis of Technological Readiness. Proceedings of 58th International Symposium ELMAR-2016. Zadar, Croatia, 2016, P. 119–122.
39. Sulema Ye. Mulsemedia Vs. Multimedia: State of the Art and Future Trends (Invited Paper). Proceedings of the 23<sup>rd</sup> IEEE International Conference on Systems, Signals and Image Processing (IWSSIP2016). Bratislava, Slovakia, 2016, P. 19–23.
40. Sulema Ye. Image Protection Method Based on Binary Operations. Proceedings of the 23<sup>rd</sup> IEEE International Conference on Systems, Signals and Image Processing IWSSIP2016. Bratislava, Slovakia, 2016, P. 295–298.
41. Sulema Ye. Haptic Interaction in Educational Applications. Proceedings of the International Conference on Interactive Mobile Communication, Technologies and Learning IMCL2015. Thessaloniki, Greece, 2015, P. 312–314.
42. Kovács P.T., Murray N., Rozinaj G., Sulema Ye., Rybárová R. Application of Immersive Technologies for Education: State of The Art. Proceedings of the International Conference on Interactive Mobile Communication, Technologies and Learning IMCL2015. Thessaloniki, Greece, 2015, P. 283–288.
43. Sulema Ye., Ebrahimi Kahou S. Statistical Approach to Image Classification. Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics AQTR 2010. Cluj-Napoca, Romania, 2010, Vol. 3, P. 1–4.

44. Sulema Ye., Ebrahimi Kahou S. Image Compression: Comparative Analysis of Basic Algorithms. Proceedings of the IEEE International Conference EWDTs'09. Moscow, Russia, 2009, P. 534–537.
45. Pester A., Sulema Ye. Multimodal Data Representation Based on Multi-Image Concept for Immersive Environments and Online Labs Development. Proceedings of the International Conference on Interactive Collaborative and Blended Learning ICBL2019. Santiago de Cuba, Cuba, 2019. P. 1–18.
46. Dychka I., Sulema O., Salenko A., Sulema Ye. Augmented Reality Application Based on Information Barcoding. Proceedings of the International Conference on Interactive Mobile Communication Technologies and Learning IMCL2019. Thessaloniki, Greece, 2019. P. 1–12.
47. Sulema Ye., Rudenko C. Fast Approximated Subsurface Scattering. Proceedings of 12<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2018. Bratislava, Slovakia, 2018. P. 1–4.
48. Sulema Ye., Bartkoviak A., Sokolovska A., Kapura A., Ohoiko M., Hurov V. Smart Room Software Based on Microsoft Kinect and Speech Synthesizing. Proceedings of 11<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2017. Bratislava, Slovakia, 2017, P. 71–74.
49. Sulema Ye., Zaichuk V., Chernyk D. Voxel-to-Polygonal Model Conversion Method. Proceedings of 11<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2017. Bratislava, Slovakia, 2017, P. 67–70.
50. Радченко Є.О., Сулема Є.С. Спосіб стеганографічного захисту графічних даних на основі схеми відповідності бітів та аналізу візуальних властивостей контейнера. Матеріали доповідей Шостої Міжнародної науково-практичної конференції з сучасних проблем



кодування, захисту й ущільнення інформації. Вінниця, Україна, 2017, С. 51–53.

51. Sulema Ye., Zaichuk V., Chernykh D. Haptic Interaction in 3D World with Use of Data Glove and Web-Camera. Proceedings of 10<sup>th</sup> International Workshop on Multimedia and Signal Processing Redžúr 2016. Slovakia, 2016, P. 45–48.
52. Sulema Y.S., Shkurat O.S. Information System for Archival Medical Images Automated Processing. Book of Abstracts of the 3rd International Conference Health Technology Management HTM-2016. Chisinau, Moldova, 2016, P. 72.
53. Sulema Ye., Valchuk Yu. Brain Tissue Image Adjustment Method. Book of Abstracts of the AMMODIT and final EUMLS Workshop. Germany, 2016, P. 31–32.
54. Sulema Ye., Shyrochyn S., Protection of Personal Graphic User Data Transmission in Computer Networks. Proceedings of International conference “Information Technology. Problems and Solutions”. Ufa, 2014, P. 7–10.
55. Сулема Є.С., Широчин С.С. Аналіз ефективності паралельної реалізації алгоритмів захисту зображень. Збірник наукових праць за матеріалами Восьмої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2014. Хмельницький, Україна, 2014, С. 335–342.
56. Сулема Є.С., Широчин С.С. Підвищення стійкості LSB-стегосистем шляхом аналізу і корекції характеристик контейнера. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2012. Хмельницький, Україна, 2012, С. 319–327.
57. Сулема Є.С., Гуренко О. Спосіб обмеження прав доступу до елементів зображення. Збірник наукових праць за матеріалами Шостої

всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2012. Хмельницький, Україна, 2012, С. 309–313.

58. Сулема Є.С., Малашихіна А. Обробка аудіо-сигналів у віртуальній мультимедійній лабораторії. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2012. Хмельницький, Україна, 2012, С. 314–318.
59. Сулема Є.С., Гнатейко В. Дослідження методів пошуку музичних файлів. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2012. Хмельницький, Україна, 2012, С. 328–332.
60. Сулема Є.С., Гуренко О. Модифікований алгоритм пошуку текстової інформації на зображеннях. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2011. Хмельницький, Україна, 2011, С. 202–206.
61. Сулема Є.С., Широчин С.С. Засоби подання інформації неграфічного характеру в картографічних зображеннях. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2011. Хмельницький, Україна, 2011, С. 207–213.
62. Сулема Є.С., Заворотній А. Спосіб розпознавання музикальних звукових образів. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» АПКТ-2011. Хмельницький, Україна, 2011, С. 214–218.
63. Сулема Є.С., Широчин С.С. Критерії пошуку оптимального розташування блоків стеганографічних даних в контейнері. Матеріали

Міжнародної науково-технічної конференції «Системний аналіз та інформаційні технології» SAIT 2011. Київ, Україна, 2011, С. 516.

64. Сулема Е., Эбрахими Кахоу С. Пиксельно-объектный способ классификации изображений. Труды десятой международной научно-практической конференции «Современные информационные и электронные технологии». Одесса, 2009. Т.1, С. 133.
65. Сулема Є.С., Ебрахімі Кахоу С. Покращення якості ущільнених зображень в автоматизованих системах відео-спостереження. Матеріали XVI Міжнародної конференції з автоматичного управління «Автоматика-2009». Чернівці, Україна, 2009, С. 54.
66. Sulema Ye., Radchuk V. Audio Processing in the Virtual Multimedia Laboratory. Proceedings of the International Conference ICL2007. Villach, Austria, 2007. P. 1–5.
67. Sulema Ye., Grebelnykova V. The Optimization of the Video Data Compression Based on Wavelet Transformation. Proceedings of the Conference on Interactive Computer Aided Learning ICL2006. Villach, Austria, 2006. P. 1–6.
68. Sulema Ye. Multimedia and Computer Technologies in e-Learning. Proceedings of the International Conference “Informational Technologies for Rural Development”. Jelgava, Latvia, 2006. P. 1–4.
69. Sulema Ye. Multimedia in Computer Networks: Technologies and Applications. Proceedings of the 10<sup>th</sup> International Netties Conference. St. Pölten, Austria, 2005. P. 1–5.
70. Sulema Ye. Using Video and Static Panoramic Images in E-Learning. Proceedings of the International Conference ICL2005. Villach, Austria, 2005. P. 1–4.
71. Sulema Ye., Sherman Eu. Secure Data Transfer in Networks. Proceedings of the Symposium on Remote Engineering Virtual Instrumentation REV 2004. Austria, 2004. P. 1–8.

72. Sulema Ye., Bhattacharya A., Murray N. Mulsemmedia Data Representation Based on Multi-Image Concept. Proceedings of the International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2019). Thessaloniki, Greece, 2019. P. 1–12.
73. Grieves M., Vickers J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. Transdisciplinary perspectives on complex systems. *Springer*, 2017. P. 85–113.
74. Grieves M. PLM-Beyond lean manufacturing. *Manufacturing Engineering*, 2003. Vol. 130, No 3. P. 23.
75. Grieves, M. Virtually indistinguishable: Systems engineering and PLM. Product lifecycle management: Towards knowledge-rich enterprises. *Springer*. 2012. P. 226–242.
76. Grieves M. Virtually perfect: Driving innovative and lean products through product lifecycle management. *Space Coast Press*, 2009. 370 p.
77. Grieve M. Product lifecycle management: The new paradigm for enterprises. *International Journal Product Development*, 2005. No 2, P. 71–84.
78. Grieves M. Product lifecycle management: Driving the next generation of lean thinking. *McGraw-Hill*, 2006. 319 p.
79. Grieves M. Virtually Intelligent Product Systems: Digital and Physical Twins. Complex Systems Engineering: Theory and Practice. *American Institute of Aeronautics and Astronautics*, 2019. P. 175-200.
80. Modeling, Simulation, Information. Technology & Processing Roadmap. Technology Area 11. *National Aeronautics and Space Administration*, 2010. 32 p.
81. NASA Technology Roadmaps. Technology area 12: Materials, structures, mechanical systems, and manufacturing. *National Aeronautics and Space Administration*, 2015. 138 p.
82. Glaessgen, E. H., Stargel, D. The digital twin paradigm for future NASA and US air force vehicles. AAIA 53rd Structures, Structural Dynamics, and

Materials Conference, Honolulu, Hawaii. 2012.

83. Eric J. T., Anthony R. I., Thomas G. E., Spottswood M. S. Reengineering aircraft structural life prediction using a digital twin. *International Journal of Aerospace Engineering*, 2011, 14. doi:10.1155/2011/154798.
84. Kritzinger W., Karner M., Traar G., Henjes J., Sihn W. Digital Twin in manufacturing: a categorical literature review and classification. *IFAC-PapersOnLine*, 2018. Vol. 51, Issue 11. P. 1016–1022.
85. Madni A. et al. Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems*, 2019. No 7. P. 1–13.
86. Lu Y., Liu C., Wang K. I-K., Huang H., Xu X. Digital Twin-driven smart manufacturing: connotation, reference model, applications and research issues. *Robotics and Computer Integrated Manufacturing*, 2020. Vol. 61, P. 1–14.
87. Alam, K. M., El Saddik, A. C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems. *IEEE Access*, 2017, Vol. 5. P. 2050–2062.
88. Redelinghuys A. J. H., Basson A. H., Kruger K. A Six-Layer Digital Twin Architecture for a Manufacturing Cell. *Studies in Computational Intelligence*, 2018. Vo. 803, P. 412–423.
89. Schleich B., Anwer N., Mathieu L., Wartzack S. Shaping the digital twin for design and production engineering. *CIRP Annals*, 2017. Vol. 66, Issue 1, P. 141–144.
90. Bevilacqua M. et al. Digital Twin Reference Model Development to Prevent Operators' Risk in Process Plants. *Sustainability*, 2020, Issue 12, Paper 1088, 17 p.
91. Ye X., Hong S. An AutomationML/OPC UA-based Industry 4.0 Solution for a Manufacturing System. *Proceedings of the IEEE 23<sup>rd</sup> International Conference on Emerging Technologies and Factory Automation ETFA*, 2018. P. 543-550.

92. Schroeder G. N., Steinmetz C., Pereira C. E., Espindola D. B. Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange. *IFAC-PapersOnLine*, 2016. Vol. 49-30, P. 12–17.
93. Lohtander M., Ahonen N., Lanz M., Ratava J., Kaakkunen J. Micro Manufacturing Unit and the Corresponding 3D-Model for the Digital Twin. *Procedia Manufacturing*, 2018. Vol. 25, P. 55–61.
94. Cai Yi, Starly B., Cohen P., Lee Y.-S. Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing. *Procedia Manufacturing*, 2017. Vol. 10, P. 1031–1042.
95. Zhu Z., Liu C., Xu X. Visualisation of the Digital Twin data in manufacturing by using Augmented Reality. *Procedia CIRP*, 2019. Vol. 81, P. 898–903.
96. Nee A. Y. C., Ong S. K., Chrysosolouris G., Mourtzis D. Augmented reality applications in design and manufacturing, *Manufacturing Technologies*, 2012. Vol. 61, No. 2, P. 657–679.
97. Novak-Marcincin J., Barna J., Janak M., Novakova-Marcincinova L. Augmented reality aided manufacturing, *Procedia Computer Science*, 2013. Vol. 25, P. 23–31.
98. Liu C., Cao S., Tse W., Xu X. Augmented Reality-assisted Intelligent Window for Cyber-Physical Machine Tools, *Journal of Manufacturing Systems*, 2017. Vol. 44, Part 2, P. 280–286.
99. Nee A.Y.C., Ong S.K. Virtual and Augmented Reality Applications in Manufacturing. *IFAC Proceedings Volumes*, 2013. Vol. 46, Issue 9, P. 15–26.
100. Segovia D., Mendoza M., Mendoza E., González E. Augmented Reality as a Tool for Production and Quality Monitoring. *Procedia Computer Science*, 2015. Vol. 75, P. 291–300.
101. Rentzos L., Papanastasiou S., Papakostas N., Chrysosolouris G. Augmented Reality for Human-based Assembly: Using Product and Process Semantics. *IFAC Proceedings Volumes*, 2013. Vol.46, Issue 15, P. 98–101.

102. Talkhestani B. A., Jazdi N., Schlögl W., Weyrich M. Consistency check to synchronize the Digital Twin of manufacturing automation based on anchor points, *Proceedings of the 51<sup>st</sup> CIRP Conference on Manufacturing Systems*, Stockholm, 2018. P. 159–164.
103. Talkhestania B. A., Jazdib N., Schlöglc W., Weyrich M. A concept in synchronization of virtual production system with real factory based on anchor-point method. *Procedia CIRP*, 2018. Vol. 67, P. 13–17.
104. Праховнік Н. А., Сізов Д. Ризики помилок під час діагностики методом цифрового двійника. *Тези конференції «Проблеми охорони праці, промислової та цивільної безпеки»*, 2019. С. 358–361.
105. Петренко Т. Г. Побудова моделі розумного вагона за технологією цифрового двійника. *Збірник наукових праць УкрДУЗТ*, 2018. Вип. 177, С. 42–43.
106. Stark R., Kind S., Neumeyer S. Innovations in digital modelling for next generation manufacturing system design. *CIRP Annals-Manufacturing Technology*, 2017. Vol. 66 (1), P. 169–172.
107. Hu L. et al. Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MT Connect. *Procedia Manufacturing*, 2018. Vol. 26, P. 1193–1203.
108. Harper E. K., Ganz Ch., Malakuti S. Digital Twin Architecture and Standards. *IIC Journal of Innovation*, November 2019. P. 1–12.
109. ISO/DIS 23247-1 Automation systems and integration. Digital Twin framework for manufacturing. Part 1: Overview and general principles <https://www.iso.org/standard/75066.htm>
110. Butterworth P. White Paper: VANTIQ Digital Twin Architecture. *VANTIQ* 2017. 17 p.
111. Rudrappa S. G. Architecture to Bridge Physical world to Virtual Digital World. 2019. <https://medium.com/@shivakumar.goniwada/architecture-to-bridge-physical-world-to-virtual-digital-world-d55ecbe93b85>

112. Wang K.-J., Lee T.-L., Hsu Y. Revolution on digital twin technology –a patent research approach. *International Journal of Advanced Manufacturing Technology*, 2020. Vol. 107, P. 4687–4704.
113. Hu L. et al. Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MTConnect. *Procedia Manufacturing*, 2018. Vol. 26, P. 1193–1203.
114. Цмоць І. Г., Батюк А. Є., Яворський А. В., Теслюк Т. В. Система моніторингу технологічних процесів «розумного підприємства». *Вісник Національного університету «Львівська політехніка» «Інформаційні системи та мережі»*, 2018. Випуск 887, С. 10–17.
115. IBM Watson IoT Platform. Product architecture. <https://www.ibm.com/support/knowledgecenter/SSQP8H/iot/overview/architecture.html>
116. Seebo Digital Twin Software. <https://www.seebo.com/digital-twin-software/>
117. Sturrock D. T. Using Commercial Software to Create a Digital Twin. Simulation for Industry 4.0. *Springer*, 2019. 288 p.
118. Smith J. S., Sturrock D. T., Kelton W. D. Simio and simulation: modeling, analysis, applications, *Simio LLC*, 2019.
119. Simio. Digital Twin. <https://www.simio.com/applications/industry-40/Digital-Twin.php>
120. Microsoft Azure Digital Twins. <https://azure.microsoft.com/en-us/services/digital-twins/>
121. STEP Tools. Digital Twin Machining. [http://www.steptools.com/blog/20171011\\_twin\\_machining/](http://www.steptools.com/blog/20171011_twin_machining/)
122. CosmoTech Enterprise Digital Twin Software Platform. <https://cosmotech.com/software-solutions/software-solution-platform>
123. SAP Digital Manufacturing Cloud. <https://help.sap.com/viewer/c86ca4026fae4cb3ba66ed751866175b/latest/en->



US/055988415fe64a7f9d43c615ab4eda5c.html

124. Autodesk. DNA for Digital Twin.  
<https://www.autodesk.com/campaigns/digital-twin>
125. Keith D. Understanding Key-Value Databases. Dataversity, 2020.  
<https://www.dataversity.net/understanding-key-value-databases/#>
126. Introduction to XML. [https://www.w3schools.com/xml/xml\\_what.asp](https://www.w3schools.com/xml/xml_what.asp)
127. Business To Manufacturing Markup Language (B2MML). <https://isa-95.com/b2mml/>
128. IEC 62264-1:2013. Enterprise-control system integration – Part 1: Models and terminology. <https://www.iso.org/standard/57308.html>
129. XML Schema Tutorial. [https://www.w3schools.com/xml/schema\\_intro.asp](https://www.w3schools.com/xml/schema_intro.asp)
130. SolidWorks. <https://www.solidworks.com/>
131. The STL Format. [https://www.fabbers.com/tech/STL\\_Format](https://www.fabbers.com/tech/STL_Format)
132. OpenGL. <https://www.opengl.org/>
133. PostgreSQL: The World's Most Advanced Open Source Relational Database. <https://www.postgresql.org/>
134. Introducing JSON. <https://www.json.org/json-en.html>
135. NoSQL Databases. <https://www.ibm.com/cloud/learn/nosql-databases>
136. Data Lake – An Overview. <https://www.yash.com/blog/data-lake-an-overview/>
137. Data Lake vs Data Warehouse. <https://luminousmen.com/post/data-lake-vs-data-warehouse>
138. Object Storage. <https://www.ibm.com/cloud/learn/object-storage>
139. Digital Twins Definition Language (DTDl). <https://github.com/Azure/opendigitaltwins-dtdl/blob/master/DTDl/v2/dtdlv2.md>
140. JSON for Linking Data. <https://json-ld.org/>
141. Use the Azure Digital Twins APIs and SDKs.  
<https://docs.microsoft.com/en-us/azure/digital-twins/how-to-use-apis-sdks>

142. Understand digital twins and their twin graph.  
<https://docs.microsoft.com/en-us/azure/digital-twins/concepts-twins-graph>
143. Introduction to Azure Data Lake Storage Gen2.  
<https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction>
144. What is Azure Synapse Analytics (formerly SQL DW)?  
<https://docs.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-overview-what-is>
145. Ghinea G., Ademoye O. A User Perspective of Olfaction-Enhanced Mulsemedia. *Proceedings of the international conference on management of emergent digital ecosystems MEDES'10*, 2010. P. 277–280.
146. Ghinea G., Andres F., Gulliver S. Multiple Sensorial Media Advances and Applications: New Developments in MulSeMedia. *Information Science Reference*, 2012. 320 p.
147. Ghinea G., Timmerer C., Lin W., Gulliver S.R. Mulsemedia: State of the Art, Perspectives, and Challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2014. Vol. 11, P. 17:1–17:23.
148. Abreu R., Mattos D., Santos J. A. F., Muchaluat-Saade D. C. Semi-automatic Synchronization of Sensory Effects in Mulsemedia Authoring Tools. *Proceedings of the 25th Brazillian Symposium on Multimedia and the Web WebMedia'19*, 2019. P. 201–208.
149. Yuan Z., Bi T., Muntean G., Ghinea G. Perceived Synchronization of Mulsemedia Services. *IEEE Transactions on Multimedia*, 2015. Vol. 17, P.957–966.
150. Mattos D. P., Muchaluat-Saade D. C., Ghinea G. An Approach for Authoring Mulsemedia Documents Based on Events. *Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC)*, 2020. P. 273–277.

151. Ademoye A. O., Murray N., Muntean G.-M., Ghinea G. Audio Masking Effect on Inter-Component Skews in Olfaction-Enhanced Multimedia Presentations. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2016. Vol. 12, No. 4, P. 51:1–51:14.
152. Montagud M. et al. MediaSync Handbook on Multimedia Synchronization. *Springer*, 2018. 679 p.
153. Nakamoto T. Human Olfactory Displays and Interfaces: Odor Sensing and Presentation. *IGI Global*, 2013. 532 p.
154. Obrist M. et al. Touch, Taste, & Smell User Interfaces: The Future of Multisensory HCI. *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2016. P. 3285–3292.
155. Kim J., Lee C.-G., Kim Y., Ryu J. Construction of a haptic-enabled broadcasting system based on the MPEG-V standard. *Signal Processing – Image Communication*, 2012. P. 1–11.
156. A. El Saddik et al., Haptics Technologies: Bringing Touch to Multimedia. *Springer*, 2011. 220 p.
157. Narumi T. Multi-sensorial Virtual Reality and Augmented Human Food Interaction. *Proceedings of the 1st Workshop on Multi-sensorial Approaches to Human-Food Interaction*, 2016. Article 1, P. 1–6.
158. Sakurai K. et al. Saltiness and Umami Suppression by Cathodal Electrical Stimulation. *Proceedings of the 1st Workshop on Multi-sensorial Approaches to Human-Food Interaction*, 2016. Article 2, P. 1–4.
159. Bruijnes M., Huisman G., Heylen D. Tasty Tech: human-food interaction and multimodal interfaces. *Proceedings of the 1st Workshop on Multi-sensorial Approaches to Human-Food Interaction*, 2016. Article 4, P. 1–6.
160. Carvalho F. R. et al. Sound-enhanced gustatory experiences and technology. *Proceedings of the 1st Workshop on Multi-sensorial Approaches to Human-Food Interaction*, 2016. Article 5, P. 1–8.
161. Mroczko-Wasowicz A. The Unity and Complexity of Flavour Perception:

- Multisensory Phenomenology and Processing in Human-Food Interactions. *Proceedings of the 1st Workshop on Multi-sensorial Approaches to Human-Food Interaction*, 2016. Article 6, P. 1–4.
162. Velasco C. et al. Multisensory Approach for the Design of Food and Drink Enhancing Sonic Systems. *Proceedings of the 1st Workshop on Multi-sensorial Approaches to Human-Food Interaction*, 2016. Article 7, P. 1–7.
  163. Kannan R., Andres F. Digital Library for Mulsemedia Content Management. *Proceedings of the international conference on management of emergent digital ecosystems MEDES'10*, 2010. P. 275-276.
  164. Amorim M.N., Saleme E.B., Assis-Neto F.R. et al. Crowdsourcing authoring of sensory effects on videos. *Multimedia Tools*, 2019. Vol. 78, P. 19201–19227.
  165. Ghinea G., et al. Towards a Reference Ontology on Mulsemedia Systems. *Proceedings of the international conference on management of emergent digital ecosystems MEDES'18*, 2018. P. 23–30.
  166. Bartocci S., Betti S., Marcone G., Tabacchiera M., Zanuccoli F., Chiari A. A novel multimedia-multisensorial 4D platform. *Proceedings of the International Annual Conference AEIT*, 2015. P. 1–6.
  167. Yoon K., Kim S.-K., Han J. J., Han S., Preda M. MPEG-V: Bridging the Virtual and Real World. *Elsevier*, 2015. 193 p.
  168. Walzl M., Timmerer C., Rainer B., Hellwagner H. Sensory effects for ambient experiences in the World Wide Web. *Multimedia Tools Application*. 2014. Vol. 70, No. 2, P. 1141–1160.
  169. Guedes A. L., Azevedo R. G., Barbosa S. D. Extending multimedia languages to support multimodal user interactions. *Multimedia Tools Application*, 2017. Vol. 76, No. 4, P. 5691–5720.
  170. Park Y., Choi B., Lee H., Sung K. A Framework of Sensory Information for 4-D Home Theater System. *Applied Mathematics & Information Sciences*, 2012. Vol. 6, No. 1S, P. 201–207.

171. Walzl M., Rainer B., Timmerer C., Hellwagner H. A toolset for the authoring, simulation, and rendering of sensory experiences. *Proceedings of the 20th ACM international conference on Multimedia*, 2012. P. 1469–1472.
172. Choi B., Lee E., Yoon K. Streaming Media with Sensory Effect. *Proceedings of the International Conference on Information Science and Applications*, 2011, P. 1–6.
173. Saleme E.B., Santos C.A.S., Ghinea G. A mulsemedia framework for delivering sensory effects to heterogeneous systems. *Multimedia Systems*, 2019. Vol. 25, P. 421–447.
174. Saleme E. B., Covaci A., Mesfin G., Santos C. A. S., Ghinea G. Mulsemedia DIY: A Survey of Devices and a Tutorial for Building Your Own Mulsemedia Environment. *ACM Computing Surveys*, 2019. Vol. 52, No. 3, P. 58:1–58:29.
175. Saleme E. B. et al. MulseOnto: a Reference Ontology to Support the Design of Mulsemedia Systems. *Journal of Universal Computer Science*, 2019. Vol. 25, No. 13, P. 1761–1786.
176. Tal I. et al. Mulsemedia in Telecommunication and Networking Education: A Novel Teaching Approach that Improves the Learning Process. *IEEE Communications Magazine*, 2019. Vol. 57, Issue 11, P. 60–66.
177. Yuan Z., Ghinea G., Muntean G.-M. Beyond Multimedia Adaptation: Quality of Experience-Aware Multi-Sensorial Media Delivery. *IEEE Transactions on Multimedia*, 2015. Vol. 17, No. 1, P. 104–117.
178. Egan D., Brennan S., Barrett J., Qiao Y., Timmerer C., Murray N. An evaluation of Heart Rate and Electrodermal Activity as an Objective QoE Evaluation method for Immersive Virtual Reality Environments. *Proceedings of the 8th International Conference on Quality of Multimedia Experience QoMEX*, 2016. P. 1–6.
179. Yuan Z., Chen S., Ghinea G., Muntean G.-M. User Quality of Experience of Mulsemedia Applications. *ACM Transactions on Multimedia Computing*,

- Communications, and Applications*, 2014. Vol. 11, P. 15:1–15:19 p.
180. Marconi C., Brovetto C., Mendez I., Perera M. Multisensorial Books: Improving Readers' Quality of Experience. *Proceedings of the 13th Latin American Conference on Learning Technologies LACLO*, 2018. P. 33–36.
  181. Mesfin G., Hussain N., Covaci A., Ghinea G. Using Eye Tracking and Heart-Rate Activity to Examine Crossmodal Correspondences QoE in Mulsemedia. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2019. Vol. 15, No. 2, P. 34:1–34:22.
  182. Mesfin G. et al. QoE of cross-modally mapped Mulsemedia: an assessment using eye gaze and heart rate. *Multimedia Tools and Applications*, 2020. Vol. 79, P. 7987–8009.
  183. Covaci A., et al. A Study on The Quality of Experience of Crossmodal Mulsemedia. *Proceedings of the international conference on management of emergent digital ecosystems MEDES'18*, 2018. P. 176–182.
  184. Danieau F. et al. HFX studio: haptic editor for full-body immersive experiences. *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, 2018. Article No. 37, P. 1–9.
  185. Danieau F. et al. H-Studio: An Authoring Tool for Adding Haptic and Motion Effects to Audiovisual Content. *Proceedings of the 26th ACM Symposium on User Interface Software and Technology UIST'13*, 2013. 2 p.
  186. Saleme E. B., Celestrini J. R., Santos C. A. S. Time Evaluation for the Integration of a Gestural Interactive Application with a Distributed Mulsemedia Platform. *Proceedings of the 8th ACM on Multimedia Systems Conference MMSys'17*, 2017. P. 308–314.
  187. Handa M., Aul G., Bajaj S. Immersive Technology – Uses, Challenges and Opportunities. *International Journal of Computing & Business Research*, 2012. P. 1–11.
  188. Manjrekar S., et al. CAVE: An Emerging Immersive Technology – A Review. *Proceedings of the 16th International Conference on Computer*

- Modelling and Simulation*, 2014. P. 130–135.
189. Stanney K. M. et al. Handbook of Virtual Environments: Design, Implementation, and Applications. *Lawrence Erlbaum Associates*. 2002. 1232 p.
  190. Timmerer C., Gelissen J., Walzl M., Hellwagner H. Interfacing with Virtual Worlds. *Alpen-Adria Universität*, 2009. P. 1–6.
  191. Megyesi Z., Barsi A., Balogh T. 3D Video Visualization on the Holovizio™ System. *Proceedings of the Conference “3DTV: The True Vision – Capture, Transmission and Display of 3D Video”*, 2008. DOI: 10.1109/3DTV.2008.4547860
  192. Covaci A. et al. 360° Multimedia: A Way to Improve Subjective QoE in 360° Videos. *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*. 2019. P. 2378–2386.
  193. 3D Camera Reviews. <http://3d-camera-review.toptenreviews.com/>
  194. Alioscopy Glasses-Free 3D Displayst. <http://www.alioscopy.com/en/3Ddisplays.php>
  195. NVIDIA 3D Vision® 2. <https://www.nvidia.com/object/product-geforce-3d-vision2-wireless-glasses-kit-us.html>
  196. Holografika. The inventor of the 3D Light Field display technology. <http://www.holografika.com/>
  197. Tangible3D. <http://www.holografika.com/Tangible3D/>
  198. HoloVizio 3D. Technology Principles. <http://www.archive.holografika.com/Technology/Technology-Principles.html>
  199. FogScreen® projection screen. <https://www.fogscreen.com/>
  200. Heliodisplay. iWave Systems. <https://www.iwavesystems.com/blog/heliodisplay/>
  201. IO2Technology: Heliodisplay/ Interactive Free-Space Display. <https://web.archive.org/web/20061119160456/http://www.io2technology.com/technology/overview.htm>

202. 3D AudioSense Spatial Sound Recording System.  
[http://www.3daudiosense.com/uploads/2/4/7/2/24724752/zylia-audiosense-parp\\_polish\\_product\\_of\\_the\\_future.pdf](http://www.3daudiosense.com/uploads/2/4/7/2/24724752/zylia-audiosense-parp_polish_product_of_the_future.pdf)
203. AudioSense HIFI In-Ear Monitors. <https://www.audiosense.cn/>
204. A3D Technology. <http://www.acoustic3d.com/products/the-emergence-as8/>
205. NewAudio Emergence AS8. <https://newaudio.com.au/>
206. Bose. Surround Sound Systems.  
[http://www.bose.com/prc.jsp?url=/shop\\_online/home\\_theater/surround\\_sound\\_systems/](http://www.bose.com/prc.jsp?url=/shop_online/home_theater/surround_sound_systems/)
207. Could ‘3D acoustic hologram generators’ put your speakers to shame?  
<http://www.digitaltrends.com/music/3d-acoustic-hologram-generator-speakers/>
208. Fujitsu Develops Prototype Haptic Sensory Tablet – Fujitsu Global  
<https://www.fujitsu.com/global/about/resources/news/press-releases/2014/0224-01.html>
209. Touch Just Got Real, Fujitsu's Haptic Sensory Tablet  
<https://www.forbes.com/sites/jenniferhicks/2014/03/15/touch-just-got-real-fujitsus-haptic-sensory-tablet/#249b81063c22>
210. VMG 30 Virtual Reality Glove – Virtual Motion Labs  
<https://www.virtualmotionlabs.com/vr-gloves/vmg-30/>
211. Silva A. J., Ramirez D. O. A., Vega V. P., Oliver O. J. P. Phantom Omni Haptic Device: Kinematic and Manipulability. *Proceedings of the IEEE Conference “Electronics, Robotics and Automotive Mechanics Conference CERMA”*, 2009. P. 193–198.
212. Butterfly Haptics. Maglev 200 product characteristics.  
<https://butterflyhaptics.com/products/system/>
213. Air-Scent. Scent Marketing Odor Control Air Care Systems.  
<https://www.airscent.com/>



214. AromaSys. <http://www.designguide.com/products/72455/AromaSys>
215. Scent, the new dimension of multimedia 4D. <https://www.exhalia.com/us/>
216. Ghinea G., Ademoye O. The sweet smell of success: Enhancing multimedia applications with olfaction. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2012. Vol. 8(1), P. 1–17.
217. ScentWave. Commercial Scent Machine. <https://scentair.com/scent-systems/scentwave>
218. Ranasinghe N.a, Lee K.-Y., Suthokumar G., Do E. Y.-L. Taste+: Digitally Enhancing Taste Sensations of Food and Beverages. *Proceedings of the 22nd ACM international conference on Multimedia (MM '14)*, 2014. P. 737–738.
219. Technology readiness levels (TRL). Horizon 2020 – Work Programme 2014-2015. General Annexes. Annex G. [https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014\\_2015/annexes/h2020-wp1415-annex-g-trl\\_en.pdf](https://ec.europa.eu/research/participants/data/ref/h2020/wp/2014_2015/annexes/h2020-wp1415-annex-g-trl_en.pdf)
220. Synchronized Multimedia Integration Language. <https://www.w3.org/TR/SMIL3/>
221. Nested Context Language. <http://www.ncl.org.br/en/inicio>
222. Nelson T. H. Complex information processing: a file structure for the complex, the changing and the indeterminate. *Proceedings of the 1965 20th national conference ACM'65*, 1965. P. 84–100.
223. ISO/IEC 23005-1:2016. Information technology – Media context and control – Part 1: Architecture. <https://www.iso.org/standard/65394.html>
224. ISO/IEC 23005-3:2013. Information technology – Media context and control – Part 3: Sensory information. <https://www.iso.org/standard/60391.html>
225. ISO/IEC 23005-5:2019. Information technology – Media context and control – Part 5: Data formats for interaction devices. <https://www.iso.org/standard/73438.html>
226. ISO/IEC 14496-33:2019. Information technology – Coding of audio-visual

- objects – Part 33: Internet video coding. <https://www.iso.org/standard/69714.html>
227. SEVino. Sensory Effect Video Annotation. <https://sourceforge.net/projects/sevino/>
228. SESim. Sensory Effect Simulator. <https://sourceforge.net/projects/sesim/>
229. SEMP. Sensory Effect Media Player. <https://sourceforge.net/projects/semmediaplayer/>
230. Booch G. et al. Object-Oriented Analysis and Design with Applications. *Addison-Wesley Professional*, 2007. 720 p.
231. Wan Z., Hudak P. Functional reactive programming from first principles. *Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation*, 2000. P. 242–252.
232. Haskell. <https://www.haskell.org/>
233. Scala. <https://www.scala-lang.org/>
234. Sousa T. B. Dataflow Programming Concept, Languages and Applications. *Doctoral Symposium on Informatics Engineering*, 2012. P. 1–12.
235. Johnston W. M., Hanna P. J. R., Millar R. J. Advances in Dataflow Programming Languages. *ACM Computing Surveys*, 2004. Vol. 36, No. 1, P. 1–34.
236. LabVIEW Community Edition. <https://www.ni.com/ru-ru/shop/labview/select-edition/labview-community-edition.html>
237. MVPL. [https://docs.microsoft.com/en-us/previous-versions/microsoft-robotics/bb483088\(v=msdn.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/microsoft-robotics/bb483088(v=msdn.10)?redirectedfrom=MSDN)
238. VHDL. <https://www.nandland.com/vhdl/tutorials/tutorial-introduction-to-vhdl-for-beginners.html>
239. Signal. <https://www.irisa.fr/espresso/polychrony/document/tutorial.pdf>
240. Morrison J. P. Flow-based Programming: A New Approach to Application Development. *J.P. Morrison Enterprises*, 2010. 352 p.
241. Paoli F., Tisato F. On the complementary nature of event-driven and time-

- driven models. *Control Engineering Practice*, 1996. Vol. 4, Issue 6, P. 847–854.
242. Raymond E. S. The Art of Unix Programming. *Addison-Wesley*, 2008. 519 p.
243. Aho A. V., Kernighan B. W., Weinberger P. J. The AWK Programming Language. *Addison-Wesley*, 1988. 210 p.
244. Robbins A. Effective Awk Programming: A User's Guide for Gnu Awk. *iUniverse*, 2019. 550 p.
245. Emerick C., Carper B., Grand C. Clojure Programming. *O'Reilly*, 2012. 598 p.
246. Latta M. A Guide to Process-oriented Programming in Elixir and OTP. <https://www.toptal.com/elixir/process-oriented-programming-elixir-and-otp>
247. Simpson J., Jacobsen C.L. Visual Process-Oriented Programming for Robotics. *Communicating Process Architectures*, 2008. P. 365–380.
248. Rozov A. S., Zyubin V. E. Process-oriented approach in MCU programming. *American Journal of Control Systems an Information Technology*, 2013. No. 1, P. 18–21.
249. Hoare C.A.R. Communicating Sequential Processes. *Programming Technologies: Communications of the ACM*, 1978. Vol. 21, No. 8, P. 666–677.
250. Sowders M. ProcessJ: A process-oriented programming language. *University of Nevada*, 2011. 132 p.
251. Welch P.H., Barnes F.R.M. Communicating Mobile Processes: Introducing occam- $\pi$ . *Lecture Notes in Computer Science*, 2005. Vol. 3525, P. 175–210.
252. Brown N. Rain: A New Concurrent Process-Oriented Programming Language. *Communicating Process Architectures*, 2006. P. 237–251.
253. Shyamasundar R. K., Ramesh S. Real time programming: languages, specification and verification. *World Scientific*, 2010, 245 p.

254. Berry G., Gonthier G. The Esterel synchronous programming language: design, semantics, implementation. *Science of Computer Programming*, 1992. Vol. 19, Issue 2, P. 87–152.
255. Berry G. Esterel v7 for the Hardware Designer. *Collège de France*, 2018. 55 p.
256. Halbwachs N., Caspi P., Raymond P., Pilaud D. The Synchronous Data Flow Programming Language LUSTRE. *Proceedings of the IEEE*, 1991. Vol. 79, No. 9, P. 1305–1320.
257. Bourke T. et al. A formally verified compiler for Lustre. *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2017. P. 586–601.
258. Washington P. et al. Interactive programming paradigm for real-time experimentation with remote living matter. *Proceedings of the National Academy of Sciences of the United States of America*, 2019. Vol.116 (12), P. 5411–5419.
259. Wang G., Cook P. R. On-the-fly Programming: Using Code as an Expressive Musical Instrument. *Proceedings of the Conference on New Interfaces for Musical Expression*, 2004. P. 138–143.
260. Wang G., Cook, P. R. ChucK: A Concurrent, On-the-fly Audio Programming Language. *Proceedings of the International Computer Music Conference*, 2003. P. 219–226.
261. Мальцев А. И. Алгебраические системы. *Наука*, 1970. 392 с.
262. Devlin K. The joy of sets. *Springer*, 1993. 193 p.
263. Kamke E. Theory of Sets. *Dover Publications*, 1950. 145 p.
264. Enderton H. B. Elements of Set Theory. *Academic Press*, 1977. 280 p.
265. Allen J.F. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 1983. P. 832–843.
266. Allen J.F., Hayes P.J. Moments and points in an interval-based temporal logic. *Computational Intelligence*, 1989. Vol. 5, Issue 3. P. 225–238.

267. Nebel B., Bürckert H.-J. Reasoning About Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra. *Journal of the Association for Computing Machinery*, 1995. Vol. 42, No. 1. P. 43–66.
268. Allen J.F., Ferguson G. Actions and Events in Interval Temporal Logic. Spatial and Temporal Reasoning. *Springer*, 1997. P. 205–245.
269. Grüninger M., Li Zh. The Time Ontology of Allen's Interval Algebra. *Proceedings of 24th International Symposium on Temporal Representation and Reasoning (TIME 2017)*, 2017. P. 16:1–16:16.
270. Schockaert S., De Cock M. Kerre E. Reasoning About Fuzzy Temporal and Spatial Information from the Web. *Intelligent Information Systems*, 2010. 594 p.
271. Bozzelli L. et al. Interval vs. Point Temporal Logic Model Checking: an Expressiveness Comparison. *ACM Transactions on Computational Logic*, 2019. Vol. 1, Article 4, 31 p.
272. Guesgen H.W., Hertzberg J., Philpott A. Towards implementing fuzzy Allen relations. *Proceedings of the Workshop on Spatial and Temporal Reasoning*, 1994.
273. Badaloni S., Giacomini M. A fuzzy extension of Allen's interval algebra. *Lecture Notes in Artificial Intelligence*, 2000. Vol. 1792. P. 155–165.
274. Schockaert S., De Cock M., Kerre E. Fuzzifying Allen's Temporal Interval Relations. *IEEE Transactions on Fuzzy Systems*, 2008. Vol. 16, No. 2. P. 517–533.
275. Bodenhofer U. A new approach to fuzzy orderings. *Tatra Mountains Mathematical Publications*, 1999. Vol. 16. P. 21–29.
276. Dubois D., Hadj A. A., Prade H. Fuzziness and uncertainty in temporal reasoning. *Journal of Universal Computer Science*, 2003. Vol. 9, P. 1168–1194.
277. Bandler W., Kohout L. J. Fuzzy relational products as a tool for analysis and synthesis of the behaviour of complex natural and artificial systems.

- Fuzzy Sets*, 1980. P. 341–367.
278. Wang X. Kerre E. Reasonable properties for the ordering fuzzy quantities (I). *Fuzzy Sets and Systems*, 2001. Vol. 118, P. 375–385.
  279. Wang X., Kerre E. Reasonable properties for the ordering fuzzy quantities (II). *Fuzzy Sets and Systems*, 2001. Vol. 118, P. 387–405.
  280. Dorohonceanu B., Marin B. A Simple Method for Comparing Fuzzy Numbers. *CiteSeer*, 2002. P. 1–6.
  281. Wang X., Kerre E. On the classification and the dependencies of the ordering methods. *Fuzzy Logic Foundations and Industrial Applications*, 1996. P. 73–88.
  282. Ramik J., Rimanek J. Inequality relation between fuzzy numbers and its use in fuzzy optimization. *Fuzzy Sets and Systems*, 1985. Vol. 16, P. 123–138.
  283. C. de Runz, Desjardin É., Herbin M., Piantoni F. A new method for the comparison of two fuzzy numbers extending Fuzzy Max Order. *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2006. P. 127–133.
  284. Wei W. Time Series Analysis. *Pearson Addison Wesley*, 2006. 614 p.
  285. Hamilton J., Hansen B. Time Series Analysis. *Econometric Theory*, 1995. Vol. 11(3), P. 625–630.
  286. Hannan E. J. Multiple Time Series. *Wiley*, 2009. 535 p.
  287. Fraenkel A. A., Bar-Hillel Y., Levy A. Foundations of Set Theory. *Elsevier*, 1973. 415 p.
  288. Петровский А. Б. Пространства множеств и мультимножеств. *Единоріал УРСС*, 2003. 248 с.
  289. Petrovsky A. B. An axiomatic approach to metrization of multiset space. *Proceedings of the 10th International Conference on Multiple Criteria Decision Making*, 1992. Vol.1. P. 381–390.
  290. Petrovsky A. B. et al. An axiomatic approach to metrization of multiset

- space. *Multiple Criteria Decision Making*, 1994. P. 129–140.
291. Petrovsky A. B. et al. Structuring techniques in multiset spaces. *Multiple Criteria Decision Making*, 1997. P. 174–184.
292. Petrovsky A. B. et al. Multiattribute sorting of qualitative objects in multiset spaces. *Multiple Criteria Decision Making in New Millennium*, 2001. P. 124–131.
293. Буй, Д. Б., Богатырёва Ю. А. Теория мультимножеств: библиография, применение в табличных базах данных. *Радіоелектронні і комп'ютерні системи*, 2010. № 7(48). С. 56–62.
294. Буй Д. Б., Гришко Ю. О. Застосування теорії мультимножин. *Радіоелектронні і комп'ютерні системи*, 2012. № 5 (57), С. 204–209
295. Кулик Б. А. Обобщенный подход к моделированию и анализу интеллектуальных систем на основе алгебры кортежей. *Труды VI Международной конференции «Идентификация систем и задачи управления» SICPRO'07*, 2007. С. 679–715.
296. Кулик Б. А., Зуенко А. А., Фридман А. Я. Алгебраический подход к интеллектуальной обработке данных и знаний. *Изд-во Политехн. ун-та*, 2010. 235 с.
297. Davies P., Maconochie I. The relationship between body temperature, heart rate and respiratory rate in children. *Emergency Medicine Journal*, 2008. Vol. 26, Issue 9. P. 641–643.
298. Ibraheem N. A., Hasan M. M., Khan R. Z., Mishra P. K. Understanding Color Models: A Review. *ARPN Journal of Science and Technology*, 2012. Vol. 2., No. 3., P. 265–275.
299. ISO/IEC 14977:1996 Information technology – Syntactic metalanguage – Extended BNF. <https://www.iso.org/standard/26153.html>
300. Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk. [https://www.researchgate.net/publication/334858202\\_Internet-of-Things\\_IoT-Based\\_Smart\\_Agriculture\\_Toward\\_Making\\_the\\_Fields\\_](https://www.researchgate.net/publication/334858202_Internet-of-Things_IoT-Based_Smart_Agriculture_Toward_Making_the_Fields_)

## Talk

301. Hill D. L., Studholme C., Hawkes D. J. Voxel similarity measures for automated image registration. *Visualization in Biomedical Computing*, 1994. Vol. 2359, No. 205. P. 1–12.
302. Laine S., Karras T. Efficient Sparse Voxel Octrees – Analysis, Extensions, and Implementation. *NVIDIA Technical Report NVR-2010-001*, 2010. 30 p.
303. Höhne K. H., Pommert A., Riemer M., Tiede U. 3D Visualization of tomographic volume data using the generalized voxel model. *Visual Computer*, 1990. Vol. 6, P. 28–36.
304. Carnero J., Diaz-Pernil D., Mari J.L., Real P. Doxelo: Towards a Software for Processing and Visualizing Topology Computations in Doxel-based 3D+t Images. *Proceedings of the 16th International Conference on Applications of Computer Algebra ACA'10*, 2010. <http://aca.aulonapress.com/index.php/aca2010/aca2010/paper/view/116>.
305. Gonzalez-Diaz R., Medrano B., Real P., Sanchez-Pelaez J. Algebraic topological analysis of time-sequence of digital images. *Lecture Notes in Computer Science*, 2005. Vol. 3718, P. 208–219.
306. Свідерський В. А. та ін. Програмне забезпечення технології портландцементу. *Строительные материалы и изделия*, 2014. № 1, С. 16-17.
307. Свідерський В. А. та ін. Програмне забезпечення технології низькотемпературних в'яжучих матеріалів. *Строительные материалы и изделия*, 2017. № 1-2, С. 22-24.
308. Blelloch G. E. Introduction to Data Compression. *Carnegie Mellon University*, 2001. 54 p.
309. Welch T. A Technique for High-Performance Data Compression. *Computer*, 1984. Vol. 17, Issue 6, P. 8–19.
310. Ziv J., Lempel A. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, 1978. Vol. 24, No. 5, P.



530–536.

311. Mehta D. P. Handbook of Data Structures and Applications. *CRC Press*, 2018. 1100 p.
312. CSV, Comma Separated Values (RFC 4180).  
<https://www.loc.gov/preservation/digital/formats/fdd/fdd000323.shtml>
313. Object Files. <http://paulbourke.net/dataformats/obj/>
314. The Virtual Reality Modeling Language Specification.  
<http://gun.teipir.gr/VRML-amgem/spec/index.html>
315. The MP3 audio file format. <https://www.ntchosting.com/encyclopedia/multimedia/mp3-audio-format/>
316. FLAC – Free Lossless Audio Codec. <https://xiph.org/flac/>
317. Ogg File Format. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000026.shtml>
318. NIfTI: Neuroimaging Informatics Technology Initiative.  
<https://nifti.nimh.nih.gov/>
319. DICOM. <https://www.dicomstandard.org/>
320. Neo4j Graph Platform. <https://neo4j.com/>
321. Amazon Neptune. <https://aws.amazon.com/neptune/>
322. Introduction to Azure Data Lake Storage Gen2.  
<https://docs.microsoft.com/en-us/azure/storage/blobs/data-lake-storage-introduction>
323. Introduction to Azure Files. <https://docs.microsoft.com/en-us/azure/storage/files/storage-files-introduction>
324. Prometheus – Monitoring system & time series database.  
<https://prometheus.io/>
325. TimescaleDB Documentation. <https://docs.timescale.com/latest/main>
326. Azure Cognitive Search – Microsoft Docs. <https://docs.microsoft.com/en-us/azure/search/search-what-is-azure-search>
327. Прэтт У. Цифровая обработка изображений. *Мир*, 1982. 312 с.

328. Гонсалес Р., Вудс Р. Цифровая обработка изображений. *Техносфера*, 2005. 1072 с.
329. Buchanan W. J., Li S., Asif R. Lightweight cryptography methods. *Journal of Cyber Security Technology*, 2017. Vol. 1, Issue 3–4, P. 187–201.
330. Ronen E., Shamir A. Extended functionality attacks on IoT devices: The case of smart lights. *Proceedings of the 2016 IEEE European symposium on security and privacy (SP'16)*, 2016. P. 3–12.
331. Dhanda S.S., Singh B., Jindal P. Lightweight Cryptography: A Solution to Secure IoT. *Wireless Personal Communications*, 2020. Vol. 112, P. 1947–1980.
332. Dutta I. K., Ghosh B., Bayoumi M. Lightweight Cryptography for Internet of Insecure Things: A Survey. *Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC2019)*, 2019, P. 0475-0481.
333. Lu C.-S. Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property. *Idea Group Inc (IGI)*, 2004. 268 p.
334. Maharjan R., Shrestha A. K., Basnet R. Image Steganography: Protection of Digital Properties against Eavesdropping. *ArXiv*, 2019. 8 p.
335. Zhenxing Q. et al. Multimedia Security: Novel Steganography and Privacy Preserving. *Security and Communication Networks*, 2018. Article ID 6390945, 2 p.
336. Gabbrielli M., Martini S. Programming Languages: Principles and Paradigms. Springer, 2010. 440 p.
337. Van Roy P., Haridi S. Concepts, Techniques, and Models of Computer Programming. *The MIT Press*, Massachusetts. 2004. 931 p.
338. Knuth D. The Art of Computer Programming. Volume 3: Sorting and Searching, Third Edition. *Addison-Wesley*, 1997. 800 p.
339. ДСТУ ISO 8601:2010 Елементи даних і формати обміну. Обмін

- інформацією. Подання дати й часу. 2010. [http://document.ua/elementi-danih-i-formati-obminu\\_-obmin-informacieyu\\_-podannj-std10547.html](http://document.ua/elementi-danih-i-formati-obminu_-obmin-informacieyu_-podannj-std10547.html)
340. Date and Time Formats. W3 Consortium. <https://www.w3.org/TR/NOTE-datetime>
341. Кораблин Ю. П. Семантика языков программирования. 1992. Издательство МЭИ. 101 с.
342. Hennessy M. The Semantics of Programming Languages. Wiley, 1990. 131 p.
343. OMG® Unified Modeling Language®, Version 2.5.1. *Object Management Group*, 2017. 796 p.
344. Rumbaugh J., Jacobson I., Booch G. The Unified Modeling Language Reference Manual. *ADDISON-WESLEY*, 1999. 568 p.
345. Code metrics values. Microsoft, 2018. <https://docs.microsoft.com/en-us/visualstudio/code-quality/code-metrics-values?view=vs-2019>
346. Maintainability Index Range and Meaning. Microsoft, 2007. <https://docs.microsoft.com/en-us/archive/blogs/codeanalysis/maintainability-index-range-and-meaning>
347. Halstead M. H., McCabe T. A. Software Complexity Measure. *IEEE Transactions on Software Engineering*, 1976. Vol. 2, No. 12, P. 308–320.
348. Medical devices. [https://www.who.int/medical\\_devices/en/](https://www.who.int/medical_devices/en/)
349. Värri A. File Exchange Format for Vital Signs and its use in digital ECG archiving. *Proceedings of 2nd Open ECG Workshop "Integration of the ECG into the EHR & Interoperability of ECG Device Systems"*, 2004. 2 p.
350. Which file format does BioSemi use? [https://www.biosemi.com/faq/file\\_format.htm](https://www.biosemi.com/faq/file_format.htm)
351. Specification for the HL7 Lab Data Interface, Oracle® Health Sciences LabPas Release 3.1. Part Number: E48677-01, 2013. 39 p.
352. Olivier B. et al. The Ultrasound File Format (UFF). *Proceedings of 2018 IEEE International Ultrasonics Symposium*, 2018. 4 p.

353. The NIFTI file format. <https://brainder.org/2012/09/23/the-nifti-file-format>.
354. The NIFTI-1 DATA FORMAT. *Data Format Working Group*, 2004. 30 p.
355. Digital Imaging and Communications in Medicine. <https://www.dicomstandard.org/>
356. CSV Files. <https://people.sc.fsu.edu/~jburkardt/data/csv/csv.html>
357. Малин А. С., Мухин В. И. Исследование систем управления. Издательский дом ГУВШЭ, 2004. 400 с.
358. International Classification of Functioning, Disability and Health. <https://www.who.int/classifications/icf/en>
359. eHealth. <https://www.who.int/ehealth/en/>
360. The Rise of mHealth Apps: A Market Snapshot. <https://liquid-state.com/mhealth-apps-market-snapshot/>
361. Hospital at Home. <https://www.johnshopkinssolutions.com/solution/hospital-at-home/>
362. Online Science Laboratories. <https://www.kent.edu/ctl/online-science-laboratories>
363. Online Labs. <https://www.golabz.eu/labs>
364. Online Labs for schools. <http://www.olabs.edu.in/>
365. Lui M. Immersive Environments. Encyclopedia of Science Education. *Springer*, 2014. 2 p.
366. Rubio-Tamayo J. L., Gertrudix B. M., García G. F. Immersive Environments and Virtual Reality: Systematic Review and Advances in Communication, Interaction and Simulation. *Multimodal Technologies Interaction*, 2017. Vol. 1 (4), 21 p.
367. Using Immersive Environments. <https://poorvucenter.yale.edu/UsingImmersiveEnvironments>

## ДОДАТОК А. Акти використання результатів дисертації

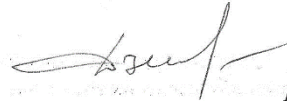


1. **Найменування пропозиції для впровадження:** математичні моделі та методи синхронізації та агрегації мультимодальних даних для формування мультиобrazу об'єкту спостереження, алгоритмічно-програмні методи оброблення мультимодальних медичних даних.
2. **Виконавець:** Сулема Є.С., к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».
3. **Джерело інформації:**
  - І. А. Дичка, Є. С. Сулема. Модель подання мультимодальних даних для комплексного опису об'єктів спостереження. Вісник Вінницького політехнічного інституту. 2020. № 1. С. 53—60.
  - Ivan Dychka, Yevgeniya Sulema, Iurii Bukhtiarov. Digital Twin Information Technology for Biomedical Data Complex Representation and Processing. Visnyk of Kherson National Technical University. 2019. No 3 (70). P. 112—119.
  - I. A. Dychka, Ye. S. Sulema. Ordering Operations in Algebraic System of Aggregates for Multi-Image Data Processing. KPI Science News. 2019. No 1. P. 15—23.
  - I. A. Dychka, Ye. S. Sulema. Logical Operations in Algebraic System of Aggregates for Multimodal Data Representation and Processing. KPI Science News. 2018. No 6. P. 44—52.
  - Yevgeniya Sulema, Ivan Dychka, Olga Sulema. Multimodal Data Representation Models for Virtual, Remote, and Mixed Laboratories Development. Lecture Notes in Networks and Systems. Springer. 2018. Vol. 47. P. 559—569.
  - Yevgeniya Sulema. ASAMPL: Programming Language for Multimedia Data Processing Based on Algebraic System of Aggregates. Advances in Intelligent Systems and Computing. Springer. 2018. Vol. 725. P. 431—442.
  - Oksana Shkurat, Yevgeniya Sulema, Viktoriya Suschuk-Sliusarenko, Andrii Dychka. Image Segmentation Method Based on Statistical Parameters of Homogeneous Data Set. Advances in Artificial Systems for Medicine and Education. Springer. 2019. Vol. 902. P. 271—281. DOI: 10.1007/978-3-030-12082-5.
  - Oksana S. Shkurat, Yevgeniya S. Sulema, Andrii I. Dychka. Complicated Shapes Estimation Method for Objects Analysis in Video Surveillance Systems. KPI Science News. 2018. No 3. P. 53—62. DOI: 10.20535/1810-0546.2018.3.136433.
  - I.A. Dychka, Ye.S. Sulema, D.A. Chernykh. Rasterization Method for Voxel Model Cutting. KPI Science News. 2018. No 2. P. 25—32.
  - Zhengbing Hu, Ivan Dychka, Yevgeniya Sulema, Yuliia Valchuk, Oksana Shkurat. Method of medical images similarity estimation based on feature analysis. International Journal of Intelligent Systems and Applications (IJISA). 2018. Vol. 10. No. 5. P. 14—22. DOI: 10.5815/ijisa.2018.05.02.
  - Zhengbing Hu, Ivan Dychka, Yevgeniya Sulema, Yevhen Radchenko. Graphical Data Steganographic Protection Method Based on Bits Correspondence Scheme. International Journal of Intelligent Systems and Applications (IJISA). 2017. Vol.9. No.8. P.34—40. DOI: 10.5815/ijisa.2017.08.04.

4. **Впроваджено:** Результати наукових досліджень впроваджені у наукову роботу кафедрою вірусології Національної медичної академії післядипломної освіти імені П.Л. Шупика
5. **Термін впровадження:** 2019
6. **Ефективність впровадження:** Впроваджено нові методи оброблення мультимодальних медичних даних у медичній вірусології, що дозволило спростити оцінку біологічної активності лікарських засобів.

**Відповідальний за впровадження:**

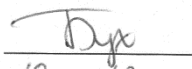
Завідувач кафедри вірусології  
Національної медичної академії  
післядипломної освіти імені П.Л. Шупика  
д.м.н., професор



I.V. Дзюблик

ЗАТВЕРДЖУЮ

Директор ТОВ «Відео Інтернет  
Технології»

 Бухтіяров Ю.В.  
«18» 12 2019р.

### А К Т

про впровадження результатів дисертаційної роботи  
Сулеми Євгенії Станіславівни

Комісія у складі: Голови – директора Бухтіярова Ю.В., керівника відділу продажів Бондар С.О. та провідного інженера Шастіна І.О. розглянула результати впровадження дисертаційної роботи «Моделі, методи та засоби подання та оброблення мультимодальних даних цифрових двійників досліджуваних об'єктів» Сулеми Є.С., що виконувалась в рамках науково-дослідної роботи «Розроблення та дослідження методів обробки, розпізнавання, захисту та зберігання медичних зображень в розподілених комп'ютерних системах» (№ держреєстрації 0117U004267) на факультеті прикладної математики КПІ ім. Ігоря Сікорського, та встановила, що внаслідок впровадження розробленої Сулемою Є.С. методики оброблення та захисту мультимодальних даних вдалось зменшити трудовитрати на розроблення відповідного програмного забезпечення у понад 2 рази.

Голова комісії:

Члени комісії:

Керівник відділу продажів

Провідний інженер

Ю.В. Бухтіяров

С.О. Бондар

І.О. Шастін



## **ДОДАТОК Б. Перелік наукових публікацій за темою дисертації та відомості про апробацію результатів**

Безпосередньо за тематикою дисертаційної роботи опубліковано 72 наукові роботи, а саме:

*Наукові публікації, в яких висвітлені основні наукові результати дисертації (26) /*

*The main scientific results of the dissertation theses were published in the next publications (26):*

*- монографії (1) / monographs (1):*

1. Mathematical Methods in Interdisciplinary Sciences / Sulema Ye., Kerre E., et al.; editor Chakraverty S. Wiley, USA, 2020. 464 p. ISBN : 9781119585640. DOI : 10.1002/9781119585640.

*- статті в наукових фахових журналах України (19) /*

*- articles in Ukrainian professional journals (19):*

2. Sulema Ye.S., Rvach D.V. Models of computation for Digital Twins data processing. *Наукові вісти КІП*, 2020. № 2, С. 74–81.
3. Sulema Ye. Multimodal data processing based on algebraic system of aggregates relations. *Radio Electronics, Computer Science, Control*, 2020. № 1, С. 169–180. [Категорія “А”].
4. Sulema Ye., Peschanskii V. Timewise data processing with programming language ASAMPL. *Вчені записки Таврійського національного університету імені В.І. Вернадського. Технічні науки*, 2020. Т.31(70), Част.1. № 1, С. 132–137.
5. Дичка І.А., Сулема Є.С. Модель подання мультимодальних даних для комплексного опису об’єктів спостереження. *Вісник Вінницького політехнічного інституту*, 2020. № 1, С. 53–60.
6. Sulema Ye., Glinskii V. Semantics and pragmatics of programming language ASAMPL. *Проблеми програмування*, 2020. № 1, С. 74–83.
7. Sulema Ye., Rudenko C. A method of artifact compensation for dual quaternion skinning and its application in digital twin models. *Вісник*



- Хмельницького національного університету. Технічні науки*, 2020. № 1, С. 12–19.
8. Сулема Є.С., Радченко Є.О. Метод стеганографічного захисту мультимедійних даних на основі процедури псевдовипадкового вбудовування. *Наукові вісті КНУ*, 2020. № 1, С. 40–47.
  9. Sulema Y.S., Los I.A. Levels-Of-Detail generation method for skeletal meshes. *Системні технології*, 2019. Т.6. № 125, С. 3–14.
  10. Сулема Є.С., Топчієв Б.С. Інтелектуальна колоризація зображень за допомогою генеративних змагальних мереж. *Системні технології*, 2019. Т.5. № 124. С. 94–103.
  11. Dychka I., Sulema Ye., Bukhtiarov Iu. Digital Twin Information Technology for Biomedical Data Complex Representation and Processing. *Вісник Херсонського національного технічного університету*, 2019. № 3 (70), С. 112–119.
  12. Dychka I.A., Sulema Ye.S. Ordering Operations in Algebraic System of Aggregates for Multi-Image Data Processing. *Наукові вісті КНУ*, 2019. № 1, С. 15–23.
  13. Dychka I.A., Sulema Ye.S. Logical Operations in Algebraic System of Aggregates for Multimodal Data Representation and Processing. *Наукові вісті КНУ*, 2018. № 6, С. 44–52.
  14. Shkurat O.S., Sulema Ye.S., Dychka A.I. Complicated Shapes Estimation Method for Objects Analysis in Video Surveillance Systems. *Наукові вісті КНУ*, 2018. № 3, С. 53–62. DOI : 10.20535/1810-0546.2018.3.136433.
  15. Dychka I.A., Sulema Ye.S., Chernykh D.A. Rasterization Method for Voxel Model Cutting. *Наукові вісті КНУ*, 2018. № 2, С. 25–32.
  16. Dychka I., Shyrochyn S., Sulema Ye. Analysis of Parallel Computations Efficiency for User's Private Multimedia Data Protection in Clouds. *Наукові вісті КНУ*, 2016. Т.1, С. 40–46.
  17. Резнік К.Ю., Сулема Є.С. Метод локалізації вушної раковини на зображенні людини у профіль. *Вісник Хмельницького національного університету*, 2014. № 4, С. 113–119.
  18. Сулема Є.С., Широчин С.С. Спосіб стеганографічного захисту даних в аудіо-файлах на основі комплементарного образу. *Наукові вісті КНУ*, 2014. Т.61, С. 80–87.
  19. Сулема Є.С., Широчин С.С. Спосіб стеганографії зображень на основі комплементарного образу. *Захист інформації*, 2013. Т.4, С. 345–353.
  20. Сулема Є.С., Широчин С.С. Спосіб стеганографії зображень з фрагментацією стегоданих та розділенням закритого ключа. *Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні*, 2012. Т.1(22), С. 64–68.

- статті в закордонних фахових виданнях, які реферуються базою Scopus (6), в тому числі:

- articles in foreign professional journals referenced by the Scopus database (6), including:

- третього квартиля (Q3) (4) / the third quartile (Q3) (4):

- 21.Sulema Ye. ASAMPL: Programming Language for Mulsemmedia Data Processing Based on Algebraic System of Aggregates. *Advances in Intelligent Systems and Computing*, 2018. Vol.725, P. 431–442. ISSN : 21945357.
- 22.Radchenko Ye., Dychka I., Sulema Ye., Suschuk-Sliusarenko V., Shkurat O. Steganographic Protection Method Based on Huffman Tree. *Advances in Intelligent Systems and Computing*, 2019. Vol.902, P. 283–292. ISSN : 21945357.
- 23.Hu Zh., Dychka I., Sulema Ye., Valchuk Yu., Shkurat O. Method of medical images similarity estimation based on feature analysis. *International Journal of Intelligent Systems and Applications (IJISA)*, 2018. Vol.10. No. 5, P. 14–22. ISSN : 20749058.
- 24.Hu Zh., Dychka I., Sulema Ye., Radchenko Ye. Graphical Data Steganographic Protection Method Based on Bits Correspondence Scheme. *International Journal of Intelligent Systems and Applications (IJISA)*, 2017. Vol.9. No.8, P.34–40. ISSN : 20749058.

- четвертого квартиля (Q4) (1) / the fourth quartile (Q4) (1):

- 25.Sulema Ye., Dychka I., Sulema O. Multimodal Data Representation Models for Virtual, Remote, and Mixed Laboratories Development. *Lecture Notes in Networks and Systems*, 2018. Vol.47, P. 559–569. ISSN : 23673389.

- реферується базою Scopus (1) /

- referenced by the Scopus database (1):

- 26.Yevgeniya Sulema, Etienne Kerre, Oksana Shkurat. Vector Image Retrieval Methods Based on Fuzzy Patterns. *International Journal of Modern Education and Computer Science (IJMECS)*. 2020. Vol.12. No.3. P. 8–16. DOI : 10.5815/ijmecs.2020.03.02.

*За результатами досліджень також опубліковано статті в інших наукових журналах України та за кордоном (7) /*

*According to the results of investigations, the articles in other scientific Ukrainian and foreign journals are also published (7):*

- 27.Sulema Ye., Papusha R. Method of Raster Line Dynamic Smoothing. *Геометричне моделювання та інформаційні технології*, 2017. № 2 (4), С. 84–89.
- 28.Сулема Є.С., Широчин С.С. Метод захисту зображень на основі шифрування палітри. *Вісник Хмельницького національного університету*, 2014. №3, С. 114–119.
- 29.Сулема Є.С., Ебрахімі Кахоу С. Адаптивний спосіб ущільнення зображень. *Вісник Хмельницького національного університету*, 2010. № 2, С. 125–131.
- 30.Павловець О.В., Сулема Є.С. Метод розпізнавання мовних одиниць на основі ADABOOST. *Вісник Хмельницького національного університету*, 2013. № 4, С. 211–222.
- 31.Sulema Ye., Ebrahimi Kahou S. Comparative Study of Image Compression Algorithms. *Radioelectronics and Informatics*, 2009. № 1. P. 1–8.
- 32.Сулема Є.С., Ебрахімі Кахоу С., Лавренчук С.С. Віртуальна мультимедійна лабораторія: особливості організації та обробки даних. *Збірник наукових праць Національного гірничого університету*, 2009. Vol.2. № 33. С. 90–96.
- 33.Sulema Ye., Cvjetkovic V. Remote Laboratory for Supporting e-Studies in Electronics. *International Journal of Online Engineering*, 2006. Vol 2, № 1, P. 1–9.

*Результати, викладені в дисертації, було апробовано на 31 науковій конференції, зокрема опубліковано 39 матеріалів конференцій /*

*The results shown in the theses were validated at 31 scientific conferences, in particular, were published in 39 materials of the conferences:*

*- що реферується базою Scopus (10) /*

*- which are referenced by the Scopus database (10):*

- 34.Sulema Ye., Amram N., Aleshchenko O., Sivak O. Quality of Experience Estimation for WebRTC-based Video Streaming. *Proceedings of the 24th*

- International Conference “European Wireless” (EW2018). Catania, Italy, 2018. P. 1–6.
- 35.Dychka I., Sulema Ye., Rudenko C. A Mathematical Model of Microsurface Normal Distribution for Specular Bidirectional Reflectance Distribution Function. Proceedings of the International Conference “Advanced Computer Information Technologies” (ACIT 2018). Ceske Budejovice, Czechia, 2018, P. 30–33.
  - 36.Sulema Ye., Rozinaj G. WebRTC-based 3D Videoconferencing System. Proceedings of 59th International Symposium ELMAR-2017. Zadar, Croatia, 2017, P. 193–196.
  - 37.Sulema Ye., Rozinaj G. Multimodal Information in Telecommunication Systems: Analysis of Technological Readiness. Proceedings of 58th International Symposium ELMAR-2016. Zadar, Croatia, 2016, P. 119–122.
  - 38.Sulema Ye. Mulsemmedia Vs. Multimedia: State of the Art and Future Trends (Invited Paper). Proceedings of the 23<sup>rd</sup> IEEE International Conference on Systems, Signals and Image Processing (IWSSIP2016). Bratislava, Slovakia, 2016, P. 19–23.
  - 39.Sulema Ye. Image Protection Method Based on Binary Operations. Proceedings of the 23<sup>rd</sup> IEEE International Conference on Systems, Signals and Image Processing (IWSSIP2016). Bratislava, Slovakia, 2016, P. 295–298.
  - 40.Sulema Ye. Haptic Interaction in Educational Applications. Proceedings of the International Conference on Interactive Mobile Communication, Technologies and Learning (IMCL2015). Thessaloniki, Greece, 2015, P. 312–314.
  - 41.Kovács P.T., Murray N., Rozinaj G., Sulema Ye., Rybárová R. Application of Immersive Technologies for Education: State of The Art. Proceedings of the International Conference on Interactive Mobile Communication, Technologies and Learning (IMCL2015). Thessaloniki, Greece, 2015, P. 283–288.
  - 42.Sulema Ye., Ebrahimi Kahou S. Statistical Approach to Image Classification. Proceedings of the IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR 2010). Cluj-Napoca, Romania, 2010, Vol. 3, P. 1–4.
  - 43.Sulema Ye., Ebrahimi Kahou S. Image Compression: Comparative Analysis of Basic Algorithms. Proceedings of the IEEE International Conference EWDTs'09. Moscow, Russia, 2009, P. 534–537.

**- інші конференції (29) / other conferences (29):**

- 44.Pester A., Sulema Ye. Multimodal Data Representation Based on Multi-Image Concept for Immersive Environments and Online Labs Development. Proceedings of the International Conference on Interactive Collaborative and Blended Learning (ICBL2019). Santiago de Cuba, Cuba, 2019. P. 1–18.
- 45.Dychka I., Sulema O., Salenko A., Sulema Ye. Augmented Reality Application Based on Information Barcoding. Proceedings of the

- International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2019). Thessaloniki, Greece, 2019. P. 1–12.
46. Sulema Ye., Rudenko C. Fast Approximated Subsurface Scattering. Proceedings of 12<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2018. Bratislava, Slovakia, 2018. P. 1–4.
47. Shkurat O., Sulema Ye., Suschuk-Sliusarenko V., Dychka A. Image Segmentation Method Based on Statistical Parameters of Homogeneous Data Set. Proceedings of International Conference of Artificial Intelligence, Medical Engineering, Education, 2018. 12 p.
48. Sulema Ye., Bartkoviak A., Sokolovska A., Kapura A., Ohoiko M., Hurov V. Smart Room Software Based on Microsoft Kinect and Speech Synthesizing. Proceedings of 11<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2017. Bratislava, Slovakia, 2017, P. 71–74.
49. Sulema Ye., Zaichuk V., Chernyk D. Voxel-to-Polygonal Model Conversion Method. Proceedings of 11<sup>th</sup> International Workshop on Multimedia Information and Communication Technologies Redžúr 2017. Bratislava, Slovakia, 2017, P. 67–70.
50. Радченко Є.О., Сулема Є.С. Спосіб стеганографічного захисту графічних даних на основі схеми відповідності бітів та аналізу візуальних властивостей контейнера. Матеріали доповідей Шостої Міжнародної науково-практичної конференції з сучасних проблем кодування, захисту й ущільнення інформації. Вінниця, Україна, 2017, С. 51–53.
51. Sulema Ye., Zaichuk V., Chernykh D. Haptic Interaction in 3D World with Use of Data Glove and Web-Camera. Proceedings of 10<sup>th</sup> International Workshop on Multimedia and Signal Processing Redžúr 2016. Slovakia, 2016, P. 45–48.
52. Sulema Y.S., Shkurat O.S. Information System for Archival Medical Images Automated Processing. Book of Abstracts of the 3rd International Conference Health Technology Management HTM-2016. Chisinau, Moldova, 2016, P. 72.
53. Sulema Ye., Valchuk Yu. Brain Tissue Image Adjustment Method. Book of Abstracts of the AMMODIT and final EUMLS Workshop. Germany, 2016, P. 31–32.
54. Sulema Ye., Shyrochyn S., Protection of Personal Graphic User Data Transmission in Computer Networks. Proceedings of International conference “Information Technology. Problems and Solutions”. Ufa, 2014, P. 7–10.
55. Сулема Є.С., Широчин С.С. Аналіз ефективності паралельної реалізації алгоритмів захисту зображень. Збірник наукових праць за матеріалами

- Восьмої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2014). Хмельницький, Україна, 2014, С. 335–342.
56. Сулема Є.С., Широчин С.С. Підвищення стійкості LSB-стегосистем шляхом аналізу і корекції характеристик контейнера. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький, Україна, 2012, С. 319–327.
57. Сулема Є.С., Гуренко О. Спосіб обмеження прав доступу до елементів зображення. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький, Україна, 2012, С. 309–313.
58. Сулема Є.С., Малашихіна А. Обробка аудіо-сигналів у віртуальній мультимедійній лабораторії. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький, Україна, 2012, С. 314–318.
59. Сулема Є.С., Гнатейко В. Дослідження методів пошуку музичних файлів. Збірник наукових праць за матеріалами Шостої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2012). Хмельницький, Україна, 2012, С. 328–332.
60. Сулема Є.С., Гуренко О. Модифікований алгоритм пошуку текстової інформації на зображеннях. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2011). Хмельницький, Україна, 2011, С. 202–206.
61. Сулема Є.С., Широчин С.С. Засоби подання інформації неграфічного характеру в картографічних зображеннях. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2011). Хмельницький, Україна, 2011, С. 207–213.
62. Сулема Є.С., Заворотній А. Спосіб розпознавання музикальних звукових образів. Збірник наукових праць за матеріалами П'ятої всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних технологій» (АПКТ-2011). Хмельницький, Україна, 2011, С. 214–218.
63. Сулема Є.С., Широчин С.С. Критерії пошуку оптимального розташування блоків стеганографічних даних в контейнері. Матеріали Міжнародної науково-технічної конференції «Системний аналіз та інформаційні технології» (SAIT 2011). Київ, Україна, 2011, С. 516.
64. Сулема Е., Эбрахими Кахоу С. Пиксельно-объектный способ классификации изображений. Труды десятой международной научно-

- практической конференции «Современные информационные и электронные технологии». Одесса, 2009. Т.1, С. 133.
- 65.Сулема Є.С., Ебрахімі Кахоу С. Покращення якості ущільнених зображень в автоматизованих системах відео-спостереження. Матеріали XVI Міжнародної конференції з автоматичного управління «Автоматика-2009». Чернівці, Україна, 2009, С. 54.
- 66.Sulema Ye., Radchuk V. Audio Processing in the Virtual Multimedia Laboratory. Proceedings of the International Conference ICL2007. Villach, Austria, 2007. P. 1–5.
- 67.Sulema Ye., Grebelnykova V. The Optimization of the Video Data Compression Based on Wavelet Transformation. Proceedings of the Conference on Interactive Computer Aided Learning ICL2006. Villach, Austria, 2006. P. 1–6.
- 68.Sulema Ye. Multimedia and Computer Technologies in e-Learning. Proceedings of the International Conference “Informational Technologies for Rural Development”. Jelgava, Latvia, 2006. P. 1–4.
- 69.Sulema Ye. Multimedia in Computer Networks: Technologies and Applications. Proceedings of the 10<sup>th</sup> International Netties Conference. St. Pölten, Austria, 2005. P. 1–5.
- 70.Sulema Ye. Using Video and Static Panoramic Images in E-Learning. Proceedings of the International Conference ICL2005. Villach, Austria, 2005. P. 1–4.
- 71.Sulema Ye., Sherman Eu. Secure Data Transfer in Networks. Proceedings of the Symposium on Remote Engineering Virtual Instrumentation REV 2004. Austria, 2004. P. 1–8.
- 72.Sulema Ye., Bhattacharya A., Murray N. Mulsemedia Data Representation Based on Multi-Image Concept. Proceedings of the International Conference on Interactive Mobile Communication Technologies and Learning (IMCL2019). Thessaloniki, Greece, 2019. P. 1–12.